FG

12

RADC-TR-76-76
Final Technical Report
March 1976

IMAGE ANALYSIS AND MODELING

Purdue University

Sponsored by
Defense Advanced Research Projects Agency
ARPA Order 2893

Approved for public release;
distribution unlimited.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

Rome Air Development Center
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public including foreign nations.

This report has been reviewed and is approved for publication.

APPROVED: David J. Brazil

DAVID J. BRAZIL, Capt, USAF
Project Engineer

Do not return this copy. Retain or destroy.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE 12		READ INSTRUCTIONS BEFORE COMPLETING FORM
14 1. REPORT NUMBER RADC-TR-76-76	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
6 4. TITLE (and Subtitle) IMAGE ANALYSIS AND MODELING.		9 5. TYPE OF REPORT & PERIOD COVERED Final Technical Report, 1 Nov 74 — 31 Oct 75
		6. PERFORMING ORG. REPORT NUMBER N/A
10 7. AUTHOR(s) Paul A. Wintz T. S. Huang		15 8. CONTRACT OR GRANT NUMBER(s) F30602-75-C-0150, ✓✓ ARPA Order-2893
9. PERFORMING ORGANIZATION NAME AND ADDRESS School of Electrical Engineering Purdue University West Lafayette IN 47907		16 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61101E AF-B8930001
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd Arlington VA 22209		11 12. REPORT DATE Mar[redacted]976 12 116p. 13. NUMBER OF PAGES 101
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Rome Air Development Center (IRRO) Griffiss AFB NY 13441		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer:
Capt David J. Brazil (IRRO)

Copies are available in DDC.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image Processing; Digital Image Processing; Image Restoration; Image Coding; Pattern Recognition; Syntactic Pattern Recognition; Image Segmentation; Texture Analysis.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report summarizes the results of the research program on Image Analysis and Modeling supported by the Defense Advanced Research Projects Agency, under Contract F30602-75-C-0150. The objective is to achieve a better understanding of image structure and to use this knowledge to develop improved image models for use in image analysis and processing tasks such as information extraction, image enhancement and restoration, and coding. The ultimate objective of this research is to form a basis for the development of technology relative to (Cont'd)

OVER

292000

military applications of machine extraction of information from aircraft and satellite imagery of the earth's surface.

TABLE OF CONTENTS

Page No.

RESEARCH SUMMARY AND OVERVIEW 2

PROJECT REPORTS

I. IMAGE SEGMENTATION

1. Image Decomposition 7
by T. S. Huang and J. W. Burnett
2. Digital Edge Restoration in Linearly Filtered Images 12
by D. P. Panda and A. C. Kak
3. Image Segmentation by Unsupervised Clustering III 24
by M. Y. Yoo and T. S. Huang
4. Texture Boundary Detection 27
by O. R. Mitchell and W. K. Chan

II. IMAGE ATTRIBUTES

5. Max-Min Measure for Image Texture Analysis 29
by O. R. Mitchell and W. A. Boyne

III. PATTERN RECOGNITION

6. Application of Outer-Product Expansions to Feature Extraction in Pictures 40
by K. Fukunaga and G. V. Sherman
7. A Branch and Bound Clustering Algorithm 43
by K. Fukunaga and P. M. Narendra

IV. PREPROCESSING

8. Image Restoration Using the Projection Method Algorithm 46
by T. S. Huang, M. Kaveh and S. Berger
9. Estimating the Impulse Response of a Degrading System 51
by Brian O'Connor and T. S. Huang

V. APPLICATIONS

10. Missile Tracking Algorithms 53
by T. S. Huang and G. Y. Tang
11. Information Extraction From γ-Rays Images 57
by G. Y. Tang
12. Satellite Imagery Noise Removal 80
by O. R. Mitchell and P. L. Chen

FACILITIES

Software 93
Hardware 96

PUBLICATIONS 97

STAFF 101

IMAGE ANALYSIS AND MODELING

Paul A. Wintz
T. S. Huang

Contractor: Purdue University
Contract Number: F30602-75-C-0150
Effective Date of Contract: 27 February 1975
Contract Expiration Date: 31 October 1975
Amount of Contract: $427,020.00
Program Code Number: 5D30
Period of work covered: Nov 74 - Oct 75

Principal Investigator: Paul A. Wintz
T. S. Huang
Phone: 317 493-3361

Project Engineer: David J. Brazil, Capt, USAF
Phone: 315 330-3175

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Capt David J. Brazil (IRRO), Griffiss AFB NY 13441.

RESEARCH SUMMARY

Since October 1, 1973, we have been supported by ARPA IPT office to carry out research in image analysis and modeling. The emphasis of our research in the past two years has been on the development of suitable mathematical models for images which are useful for image processing tasks such as efficient coding, enhancement, recognition, and information extraction. Our accomplishments have been recorded in detail in our progress reports. We summarize here some of the highlights.

A. Preprocessing

We have obtained significant results in both image restoration and efficient coding.

1. Image phase [1].

Although it is well known that the phase of the Fourier transform of an image is generally more important than the magnitude, most past work on two-dimensional digital filter design concentrated on magnitude specification only. We demonstrated that the phase accuracy of image processing filters are extremely important. Even if the desired filter has linear phase, failure to specify it may lead to disaster. We also developed methods of designing two-dimensional digital filters which specify both the magnitude and the phase.

2. Recursive estimation.

By modeling images as two-dimensional random fields, it becomes feasible to apply the Kalman formulation of recursive estimation to image restoration. However, in trying to derive the optimum two-dimensional estimator, one encounters fundamental mathematical difficulties. We have nevertheless developed several suboptimum estimators which in practice perform almost as well as the optimum.

3. Iterative image restoration [2,3,4].

Many image degradations can be approximated by linear models. Then image restoration on the computer becomes the problem of solving a set of linear algebraic equations. Because of image noise, conventional solution methods are unusable. We have developed an iterative method (the projection method) for doing the restoration which offers a tradeoff between noise and image sharpness. Compared to the singular value decomposition method studied at Purdue and elsewhere, this iterative method offers a similar performance with drastically reduced computation requirements.

4. Three-dimensional reconstruction [5].

We have studied several methods of reconstructing three-dimensional structures from two-dimensional x-ray pictures. Specifically, we have investigated the effects of quantization, beam divergence, and unknown beam strength. It was found that 64 levels of quantization is sufficient, that a beam divergence of up to 10° can be tolerated, and that an unknown beam strength introduces a ring-like structure to the reconstruction.

5. Error-free DPCM codes for ERTS Imagery [6,7].

The difference statistics of ERTS imagery have been measured; and based on these several easily implementable classes of DPCM codes have been developed. They reduce the bit rate from 8 bits per picture element to about 4 bits per picture element. A slightly more complicated adaptive code reduces the bit rate to about 2.5 bits per picture element. These codes are error-free in the sense that they do not introduce any distortion to the images.

B. Image Segmentation

We have taken two approaches to image segmentation: edge extraction, and region growing.

1. Edge extraction [8].

We have developed several improved local operators for edge extraction. They work quite well even in a noisy and blurred environment. These operators have been applied to change detection of missile imagery with excellent results.

2. Region growing [9].

We have developed a computer algorithm called BLOB which segments an image into regions so that points in the same region have similar characteristics. The BLOB algorithm was used to increase the accuracy of classifying multispectral ERTS data. These data had been classified point by point using spectral signature. By the application of BLOB, regional classification became possible. This not only increased the classification accuracy (by about 5% on the average) but also reduced the classification time (by a factor of of about 30:1).

3. Image decomposition [8].

Image decomposition can be considered as a generalization of image segmentation. In the latter, an image is segmented into non-overlapping regions: while in the former, an image is decomposed into a sum of generally overlapping images. For example, an image might be decomposed into the sum of a low-spatial-frequency image and a high-spatial-frequency image. Many image processing tasks are facilitated if the image is decomposed into simpler components and each component is handled according to its own characteristics. We have been developing algorithms to decompose an image into three components: edges, background (slowly-changing), and textures. This idea of image decomposition has been applied to image noise reduction. Applying a Wiener filter to a noisy image reduces the noise but also blurs the edges of the objects in the image. By treating the edges separately, one is able to reduce the noise and in the same time retain edge sharpness.

C. Pattern Classification

1. Texture analysis [10].

We have developed a set of image texture descriptors based on extrema along scan lines. These texture descriptors are very easy to measure, yet perform as well as or even better than conventional (tedious to calculate) descriptors (such as the spatial-dependence matrix) in most pattern recognition tasks.

2. Optimum feature subset selection algorithms [11].

Patterns to be classified by a recognition system are generally characterized by a set of measurements or features. Often, the dimensionality of this feature space is too large for efficient and reliable application of existing classification techniques. The feature extraction problem is then, to reduce the dimensionality of the feature space, without significantly affecting the discriminatory capacity of the feature set. One approach to feature extraction is to select a smaller subset of m features of the set of n original features ($m < n$). Exhaustive search is computationally unfeasible even for modest values of n and m. Using a branch and bound approach, we have developed algorithms which are efficient and guaranteed to be optimal. These algorithms were applied to choose the best subset of 12 out of 24 channels. There are 2,704,150 candidate solutions. The algorithms obtained the best subset with the computational effort equivalent to computing the criterion for 6000 subsets. The savings are indeed substantial.

References

[1] T. S. Huang, J. Burnett, and A. Deczky, "The importance of phase in image processing filters," to appear in IEEE Trans. on Acoustics, Speech, and Signal Processing.

[2] T. S. Huang, B. Barker, and S. Berger, "Iterative image restoration," Applied Optics, May 1975.

[3] T. S. Huang, M. Kaveh, and S. Berger, "Some further results in Iterative Image Restoration," to be presented at the Annual meeting of the Optical Society of America, October 21-24, 1975; Boston, Mass.

[4] T. S. Huang, and P. M. Narendra, "Image restoration by singular value decomposition," to appear in Applied Optics, September 1975.

[5] A. Kak, C. Jakowatz, N. Baily, and R. Keller, "Computerized tomography using video recorded flouroscopic images," TR-EE 75-22, School of Electrical Engineering, Purdue University, West Lafayette, Indiana, June, 1975.

[6] J. R. Duan, and P. A. Wintz, "Information-preserving coding for multispectral scanner data," TR-EE 74-15, School of Electrical Engineering, Purdue University, West Lafayette, IN 1974.

[7] T. S. Huang, "Some easily implementable suboptimum codes," ICC Convention Record, June 16-18, 1975; San Francisco, CA.

[8] T. S. Huang, and G. Y. Tang, "Application of edge detection to image enhancement," presented at the 5th Annual Symp. on Automatic Imagery Pattern Recognition, April 17-18, 1975, University of Maryland, College Park, Maryland.

[9] J. Gupta, and P. A. Wintz, "A boundary finding algorithm and its applications," IEEE Trans. on Circuits and Systems, March 1975.

[10] O. R. Mitchell, and C. Meyers, "Texture Analysis," presented at the 5th Annual Symp. on Automatic Imagery Pattern Recognition, April 17-18, 1975; University of Maryland, College Park, Maryland.

[11] P. Narendra, and K. Fukunaga, "Optimum feature subset selection algorithms," submitted to IEEE Trans. on Computers.

IMAGE DECOMPOSITION

T. S. Huang and J. W. Burnett

I. Introduction

In previous reports [1], [2] we have modeled image scan lines as Markov jump processes. This model led to nonlinear noise reduction and image segmentation algorithms that are superior to linear techniques currently in use.

The recursive calculation of a conditional probability involving the boundary component of the scan line was the key to the nonlinear algorithms. Once this conditional probability had been calculated the scan line could be segmented using a maximum likelihood approach or noise could be reduced with either maximum liklihood or minimum mean square error estimators. Further, since the conditional probability obeyed a recursive relationship the execution time and memory requirements of these algorithms were kept to a minimum.

In this report we start to extend the one dimensional results to two dimensions. The first step in this extension is the recursive estimation of a two dimensional constant.

II. Derivation

Assume we have the observations

$$(1) \quad R(x,y) = Ixy + BW(x,y)$$

where I is one of K possible values $a_1, a_2, \ldots, a_K$ and W(x,y) is a two dimensional Wiener process (the integral of white Gaussian noise.)

Equation (1) is sometimes written as

$$(1') \quad dR = Idxdy + BdW \qquad \text{or}$$

$$(1'') \quad Z = \frac{\partial^2 R}{\partial x \partial y} = I + BN(x,y)$$

where $N(x,y)$ is white Gaussian noise.

Let $S_{rn} = \frac{rx}{n}$ $\quad t_{qn} = \frac{qy}{n}$

$r,q = 1, 2, 3, \ldots, n$

$n = 1, 2, 3, \ldots$

Define $\eta_{rqn} = R(S_{rn},t_{qn}) + R(S_{r-1n},t_{q-1n}) - R(S_{rn},t_{q-1n}) - R(S_{r-1n},t_{qn})$

From (1)

$$\eta_{rqn} = \frac{Ixy}{n^2} + B\Delta W_{rqn} \qquad r,q = 1, 2, \ldots, n$$

where

$$\Delta W_{rqn} = W(S_{rn},t_{qn}) - W(S_{r-1n},t_{qn}) - W(S_{rn},t_{q-1n}) + W(S_{r-1n},t_{q-1n})$$

Now ΔW_{rqn} is a linear combination of Gaussian random variables and hence Gaussian. Further, since W is an independent area process ΔW_{rqn} is independent of ΔW_{abn} for $a \neq r$, $b \neq q$. Thus

$$\eta_{rqn} - \frac{IXY}{n^2} \qquad r,q = 1, 2, \ldots, n$$

are independent zero mean Gaussian random variables with variance $\frac{B^2xy}{n^2}$.

Let $P_j^{(n)} = P(I = a_j | R(S_{rn},t_{qn}),\ r,q = 1, 2, \ldots, n$

= conditional probability the random field I has value a_j given the observations R upto the point x,y.

Note that η_{rqn} is obtained from R by an operation that (except for some boundary conditions) is nonsingular. Thus if we define $R(x,y) = 0$ for $x \leq 0$ or $y \leq 0$ then

$$P(I = aj \mid R(S_{rn}, t_{qn})) = P(I = a_j \mid \eta_{rqn})$$

From Bayes rule we have

$$P_j^{(n)}(x,y) = \frac{P_j(0,0)\,\exp[-\frac{n^2}{2B^2 xy}\sum_{r,q=1}^{n}(\eta_{rqn} - \frac{a_j xy}{n^2})^2]}{\sum_{\ell=1}^{K} P_\ell(0,0)\,\exp[-\frac{n^2}{2B^2 xy}\sum_{r,q=1}^{n}(\eta_{rqn} - \frac{a_\ell xy}{n^2})^2]} \qquad (2)$$

where $P_j(0,0)$ is the a-priori probability that $I = aj$.

Equation (2) can be rewritten as

$$P_j^{(n)}(x,y) = \frac{P_j(0,0)\,\exp[\frac{1}{B^2}\sum_{r,q=1}^{n}\eta_{rqn}a_j - \frac{a_j^2 xy}{2B^2}]}{\sum_{\ell} P_\ell(0,0)\,\exp[\frac{1}{B^2}\sum_{r,q=1}^{n}\eta_{rqn}a_\ell - \frac{a_\ell^2 xy}{2B^2}]}$$

By definition of the two dimensional stochastic integral [3],

$$\lim_{n\to\infty} \frac{1}{B^2} a_j \sum_{r,q=1}^{n} \eta_{rqn} = \frac{a_j}{B^2} \int_o^x \int_o^x dR(u,v)$$

$$= \frac{a_j}{B^2} R(x,y)$$

Since exp (·) is a continuous function

$$\lim_{n\to\infty} P_j^n(x,y) = P_j(x,y) = \frac{P_j(0,0)\,\exp[B^{-2}a_j R(x,y) - \frac{a_j^2 xy}{2B^2}]}{\sum_{\ell=1}^{K} P_\ell(0,0)\,\exp[\frac{a_\ell}{B^2}R(x,y) - \frac{a_\ell^2 xy}{2B^2}]}$$

Each P_j is of the form

$$\frac{f_j(x,y)}{\sum_\ell f_\ell(x,y)} \quad \text{where } f_\ell = P_\ell(0,0)\,\exp[B^{-2}a_\ell R(x,y) - \frac{a_\ell^2 xy}{2B^2}]$$

Using a result of Wong [3], [4] it can be shown that each f_j obeys a recursive relationship

$$df_j(x,y) = B^{-1}a_jf_j(x,y)dR(x,y) +$$

$$B^{-2}a_j^2f_j(x,y)\int_0^x\int_0^y dR(\xi_1,y)dR(x,\xi_2) \qquad (4)$$

Thus an incremental change in each f_j can be calculated by knowing the "present value" of f_j and the present value of the observation (see Fig. 1).

III. Conclusions

Equation (4) is significant in that it gives a two dimensional recursive method for calculating a conditional probability. From (4) we can recursively calculate each f_j and hence each P_j. While this result may not find widespread application it nonetheless represents the first step in deriving a recursive expression for a conditional probability when the random field I (1) is a Markov jump process.

REFERENCES

[1] J. Burnett and T. S. Huang, "Image Decomposition," in Image Analysis and Modeling Interim Report, Feb. 1 - Apr. 30, 1975.

[2] J. Burnett and T. S. Huang, "Image Decomposition," in Image Analysis and Modeling Interim Report, 1975.

[3] E. Wong, "A Liklihood Ratio Formula for Two Dimensional Random Fields," IEEE Trans. Info. Theory, Vol. IT-20, July 1974.

[4] E. Wong, "Recursive Filtering for Two Dimensional Random Fields," IEEE Trans. Info. Theory, Vol. IT-21, January 1975.

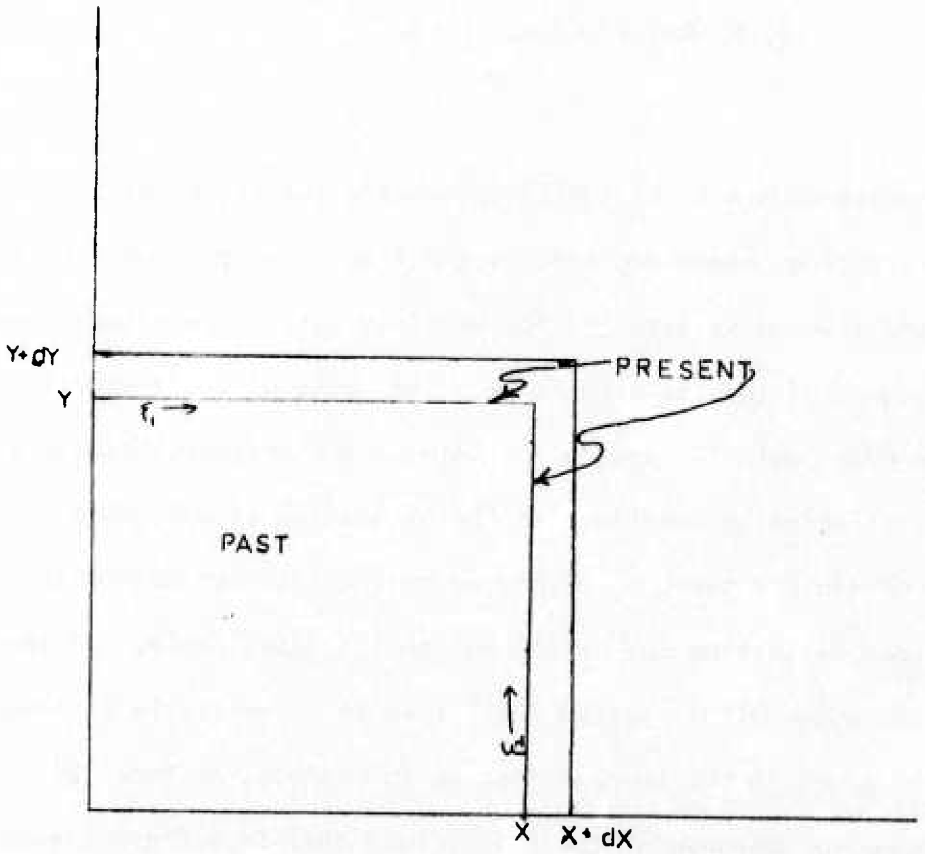


Figure 1 Recursive Calculation of df(x,y) from past values of f(x,y) and present values of dR(x,y)

DIGITAL EDGE RESTORATION IN LINEARLY FILTERED IMAGES

D. P. Panda and A. C. Kak

1. INTRODUCTION

Linear minimum means square error (MMSE) estimation techniques have commonly been used in restoring images degraded by the imaging system and observed in the presence of additive noise (see [1] for brief survey). These techniques often distort the "edges" of the restored image. The cause of the distortion is the fact that the edges mark the boundaries between two adjacent "clusters" of different local statistical properties and the estimation of a picture element (pixel) in one cluster nearby. Depending on the intended use of the restored image the edge distortion may or may not be too undesirable. If the image is meant for the human visual system (HVS) then it is desirable to have sharp and undistorted edges in the image. This is so because, as Hunt [2] pointed out, the frequency response of the HVS is such that it differentiates the lower spatial frequencies and amplifies the higher spatial frequencies which amounts to emphasizing the edge contents of the image. When a linear MMSE restoration filter does not have any edge preserving constraint or measure it shall be necessary to find a post-processing technique which would enhance the edges in the restored image (the term restored image is used here to mean the output of the linear MMSE restoration filter). Many frequency domain methods are already available for edge enhancement. One such frequency domain technique is due to Schreiber [3] in which he subtracted from the image a weighted low-pass version of it to increase the relative amplitudes of the high spatial frequency components and restore some of the sharpness. Frequency domain operations may be preferred when being implemented optically. But in digital implementation spatial domain operations are expected to be more attractive than frequency domain operation from the storage and computation point of view. We present here a spatial domain post-processor for the enhancement of the edges in the images restored linearly by the MMSE criterion.

II. EDGE DISTORTION IN LINEARLY FILTERED IMAGES

There are several ways the edges in an image can be modelled. Each of the many edge detection schemes devised by various researchers is based implicitly on a specific model of the edges. In terms of power spectral density the edges contribute to the relatively higher, spatial frequency components of an image. In some spatial domain edge detection methods an edge is defined as the set of pixels that mark the transition of the local mean grey level from one value to another, while in other methods the change in the variance of the grey levels is also taken into account. Whether it is the first, the second, or the higher order statistics that are used for edge detection it is obvious that the edges are the boundaries of regions that contain different local statistical properties. When these local properties are used as features in a clustering procedure the above mentioned regions cluster together in the feature space. These clusters in the feature space are separated from one another by the edges in the image space and, in general, correspond to different objects in the object space (see Fig. 1). Researchers in the area of remote-sensing make use of this concept in the spatial clustering of multi-image data and scene classification (see, for example, Haralick and Dinstein [5] and Gupta and Wintz [4]). We shall use here the terms edges, "cluster boundaries", and boundaries interchangeably.

When a linear estimator is used for filtering, the local properties of an estimated pixel are affected by the local properties of all the pixels included in the observation set for estimating that pixel. Therefore, in the neighborhood of the boundaries of two clusters the estimated pixels will have local properties depending on the local properties of both the clusters rather than just the cluster to which the pixel belongs.

Definitions

Let s(p,q) be the grey level of the Image at the point (p,q). Suppose the image is scanned in certain direction θ, $0 \leq \theta \leq \pi$. In any particular scan line we define a run R as the set of consecutive pixels such that the grey level of those pixels is monotomically increasing or decreasing. If (p_1,q_1) and (p_2,q_2) are the two end points of the run then we define the contrast of the run as

$$C(R) = |s(p_1,q_1) - s(p_2,q_2)|/[s(p_1,q_1) + s(p_2,q_2)]$$

and for any point (p,q) in the run we define sharpness

$$\Psi_\theta(p,q) = \frac{C(R)}{((p_1-p_2)^2 + (q_1-q_2)^2)^{1/2}}$$

We define the sharpness error criterion between an image and its estimate as

$$e(p,q) = \sum_{\theta=\theta_1}^{\theta_k} [\Psi_\theta(p,q) - \hat{\Psi}_\theta(p,q)]^2$$

where ^ indicates the value in the estimated image, θ_1, θ_2, ..., θ_k are k different directions along which the images are to be scanned for computing the error.

III. EDGE RESTORATION

Since the effect of pixels from one cluster on the pixels of a neighboring cluster near the boundary Is the cause of the edge distortion, If we introduce in the linear filter nonlinearity near the boundary such that an estimated pixel of a cluster Is a linear combination of pixels of only that cluster then we should expect a great reduction in the distortion. But if no such attempt Is made to prevent the edge distortion in the linear filter then a post-processor is needed to correct the possible distortion in the image. The post-processor presented here does this correction in the following steps. First it detects in the output of the linear filter the boundaries between adjacent clusters. Then it changes the grey levels and the gradients of the pixels in those regions in such a fashion that the sharpness of the boundary is increased. If needed, it smooths out further the regions that do not constitute the boundary.

Edge Detection

Let $s(p_1,q_1)$ and $s(p_2,q_2)$ be the grey levels at pixels (p_1,q_1) and (p_2,q_2) in adjacent clusters 1 and 2, respectively. We will assume that $s(p_1,q_1)$ and $s(p_2,q_2)$, respectively, are Gaussian with means μ_1 and μ_2 and variances σ_1^2 and σ_2^2. Also, for tractability in deriving the boundary detection algorithm we will replace the expression for contrast of a run R by the following

$$\text{local contrast } C(R) = (s(k,\ell) - s(i,j))/2\mu \qquad (1)$$

where μ is the average grey level of the region, and (k,ℓ) and (i,j) are the maximum and the minimum end points, respectively, of the run R. When the run R is across a boundary those points will not be in the same cluster. Let

$$D(R) = s(k,\ell) - s(i,j)$$

For a run R across a boundary the conditional probability density of D given that a boundary has been detected is given by

$$p_D(d|B) \approx N(\mu_b,\sigma_b{}^2) \qquad (2)$$

where B is the event that an edge has been detected. μ_b and $\sigma_b{}^2$ are the mean and the variance respectively of the Gaussian density function. These parameters are given by

$$\begin{aligned} \mu_b &= \mu_2 - \mu_1 \\ \sigma_b^2 &= \alpha_1^2 + \alpha_2^2 \end{aligned} \tag{3}$$

If instead the pixels (k,ℓ) and (i,j) belonged to the same cluster, say cluster 1, i.e., (k,ℓ) and (i,j) are two pixels at the opposite ends of a run of monotonically increasing or decreasing grey levels but the region between them is not a cluster boundary, then

$$p_D(d|C) \simeq N(\mu_c,\sigma_c^2) \tag{4}$$

where C denotes the event that the run is inside a cluster. The parameters of the Gaussian density function $N(\mu_c,\sigma_c^2)$ are given by

$$\mu_c = 0$$

and

$$\sigma_c^2 = 2\sigma_1^2 \tag{5}$$

If it is assumed that in the output of the linear filter the runs are of approximately equal length then given a particular value of D(R) whether or not the run R is across a cluster boundary is now a pattern recognition problem. If P(B) and P(C), respectively, are the prior probabilities of the existence of a boundary and a cluster in a particular region then the Bayesian solution to the pattern recognition problem is

$$\begin{aligned} \frac{p(d|B)\ P(B)}{p(d|C)\ P(C)} &> 1 \quad \text{implies B} \\ &= 1 \quad \text{implies B or C} \\ &< 1 \quad \text{implies C} \end{aligned} \tag{6}$$

Using appropriate Gaussian densities for $p(d|B)$ and $p(d|C)$ we get the discriminant functions,

$$g_b(d) = -\frac{1}{2}\frac{d^2}{\sigma_b^2} + \frac{d\mu_b}{\sigma_b^2} - \frac{1}{2}\left(\frac{\mu_b^2}{\sigma_b^2} + \log \sigma_b^2\right) + \log P(B)$$

$$g_c(d) = -\frac{1}{2}\frac{d^2}{\sigma_c^2} + \frac{d\mu_c}{\sigma_c^2} - \frac{1}{2}\left(\frac{\mu_c^2}{\sigma_c^2} + \log \sigma_c^2\right) + \log P(C) \qquad (7)$$

In terms of the discriminant function, (6) can be expressed as

$$g_b(d) - g_c(d) > 0 \quad \text{implies edge}$$

$$< 0 \quad \text{implies no edge} \qquad (8)$$

There exists a tradeoff between the accuracy of the detection of boundary and the speed and the cost of computation involved. Where as using both the first and the second order statistics is expected to give better performance cost. Often just the first order statistics gives satisfactory results with less cost. Hence, we take the variance σ_1^2 and σ_2^2 to be equal $(= \sigma^2)$ thus reducing the complexity of the discriminant function as shown below

$$\sigma_b^2 = \sigma_c^2 = 2\sigma^2$$

Therefore

$$g_B(d) = \frac{d\mu_b}{2\sigma^2} - \frac{\mu_b^2}{4\sigma^2} + \log P(b)$$

$$g_C(b) = \frac{d\mu_c}{2\sigma^2} - \frac{\mu_c^2}{4\sigma^2} + \log P(c)$$

The decision rule now is

$$d > t \rightarrow \text{edge}$$

$$< t \rightarrow \text{no edge}$$

where the threshold t is

$$t = \frac{\dfrac{\mu_b^2 - \mu_c^2}{4\sigma^2} + \log \dfrac{P(c)}{P(b)}}{\dfrac{\mu_b - \mu_c}{\substack{2 \\ 2\sigma}}}$$

The probability of error of the classifier is decreased, at the cost of a little computational complexity, by changing the decision rule as follows:

$$g_B(d) - g_C(d) > t_b \rightarrow \text{boundary}$$

$$< t_c \rightarrow \text{no boundary}$$

$$\text{otherwise undecided}$$

When undecided a new run of monotonically increasing or decreasing grey levels is added to the previous run. This process is continued till a boundary is detected. The width of the boundary is the cumulative length of all runs used in making the decision. Since the edge width in the output of a linear filter does not distort without limit an empirically determined threshold can be set for the cumulative run length. If at any time in a decision making the cumulative run length meets the threshold "no boundary" is decided.

Boundary Enhancement and Cluster Smoothing

The sharpness of the edge can be increased either by increasing local contrast or by reducing the boundary width. To increase the contrast by a small fraction we must increase or decrease the grey levels in the boundary region to a great extent which will be "out of step" with the rest of the pixels in the cluster, thereby creating artifacts. For example, if

$$s_2(k,\ell) = 75$$

$$s_1(i,j) = 25$$

$$\text{contrast} = \frac{75-25}{75+25} = 1/2$$

Increasing $s_2(k,\ell)$ 5 times to 375 or decreasing $s_1(i,j)$ 5 times to 5 gives

$$\text{contrast} = \frac{375-25}{375+25} = \frac{350}{400} = 7/8$$

or

$$\text{contrast} = \frac{75-5}{75+5} = \frac{70}{80} = 7/8$$

which is even less than a twofold improvement. But decreasing the boundary width to, say, 1 gives a very large order of improvement in the sharpness. This improvement is proportional to run length. The worse the edge distortion is the larger is the run length and consequently the higher is the improvement in the sharpness of the edge.

The boundary width is reduced by a nonlinear mapping of the grey levels by which the grey levels above the midpoint get mapped to $s_2(k,\ell)$ and those below the midpoint get mapped to $s_1(i,j)$.

The next step after enhancement of the edges is the smoothing of the interior of the clusters. This smoothing may or may not be necessary depending on the residual noise present in the linearly filtered image. If, for example, the linear filter, such as the one by Habibi [6], is the kind that used the image correlation coefficients as its parameters then the amount of residual noise in a cluster in the filtered image might depend on how well the correlation coefficients used in the filter design match that of the cluster under consideration. Usually the correlation coefficients have to be estimated from one or a few sample images and unless the restoration filter is adaptive it will be impossible for it to match all the clusters encountered. When needed, the smoothing is done in the post-processor by a simple first order digital recursive filter. The pole-zero locations of the filter may be controlled to get the desired amount of smoothing.

REFERENCES

[1] T. S. Huang, W. F. Schreiber, and O. J. Tretiak, "Image Processing," Proc. IEEE, Vol. 59, No. 11, pp. 1586-1609, Nov. 1971.

[2] B. R. Hunt, "Digital Image Processing," Proc. IEEE, Vol. 63, No. 4, pp. 693-708, April 1975.

[3] W. F. Schreiber, "Wirephoto Quality Improvement by Unsharp Masking," Pattern Recognition, Vol. 2, pp. 117-121, 1970.

[4] J. N. Gupta and P. A. Wintz, "A Boundary Finding Algorithm and its Applications," IEEE Trans. on Ckt. and Syst., Vol. CAS-22, No. 4, pp. 351-362, April 1975.

[5] R. M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi-Image Data," IEEE Trans. on Ckt. and Syst., Vol. CAS-22, No. 5, pp. 440-450, May 1975.

[6] A. Habibi, "Two-Dimensional Bayesian Estimates of Images," Proc. IEEE, Vol. 60, No. 7, pp. 878-883, July 1972.

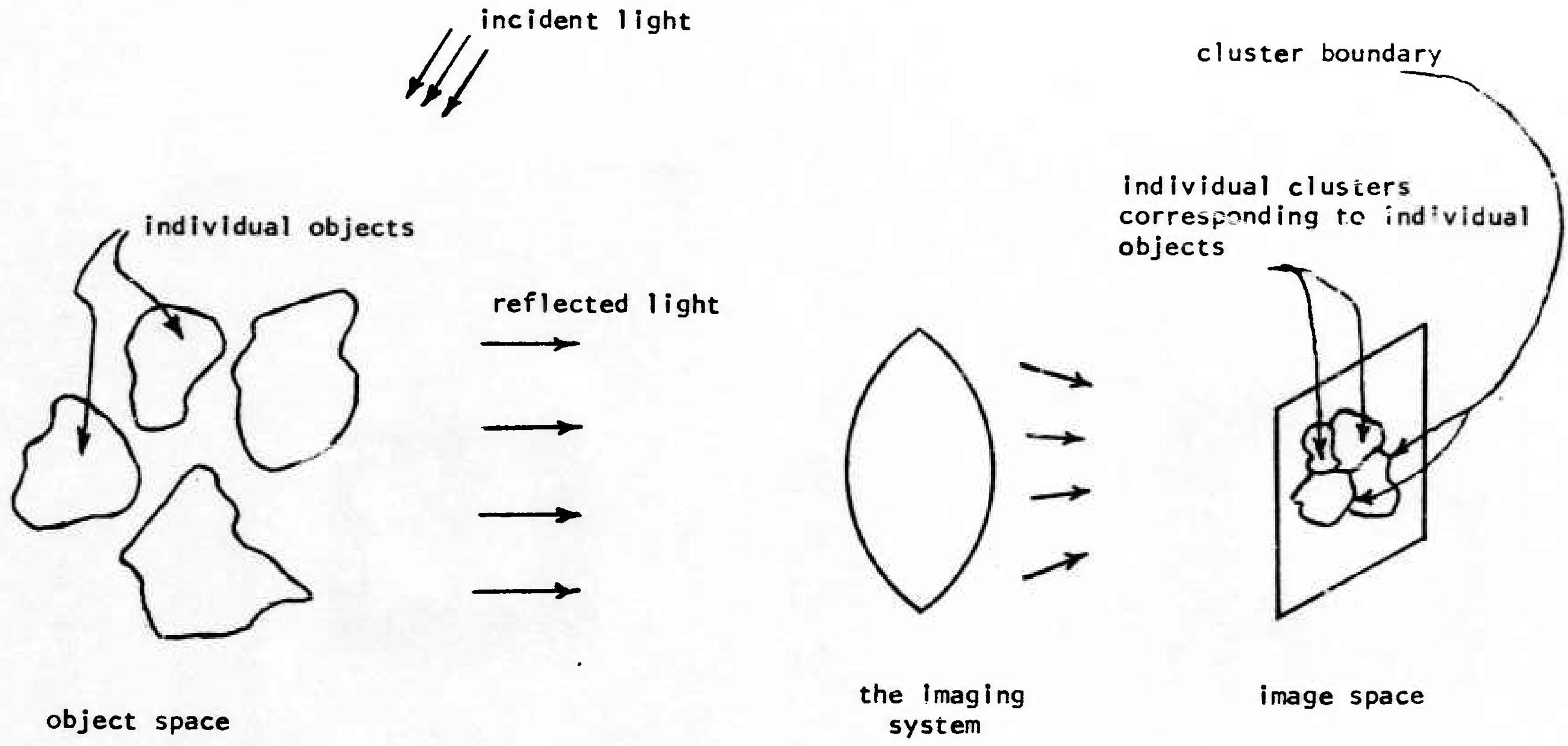


Fig. 1 Transformation from the Object Space to the Image Space

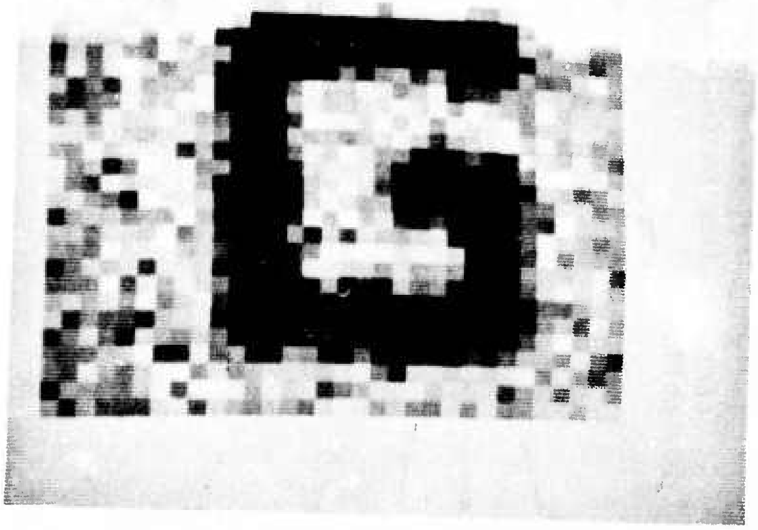


Figure 2 A test image blurred by a Gaussian shape point spread function and observed in the presence of additive observation noise

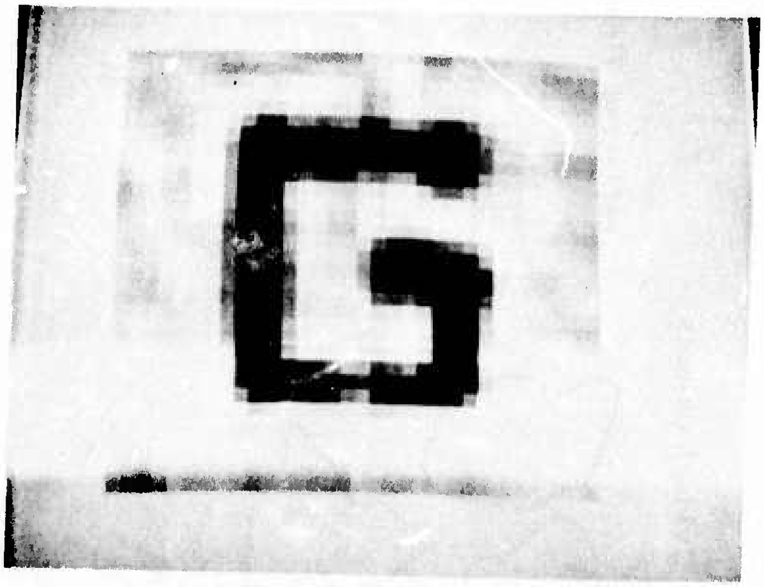


Figure 3 The image of Fig. 2 filtered by a linear filter.

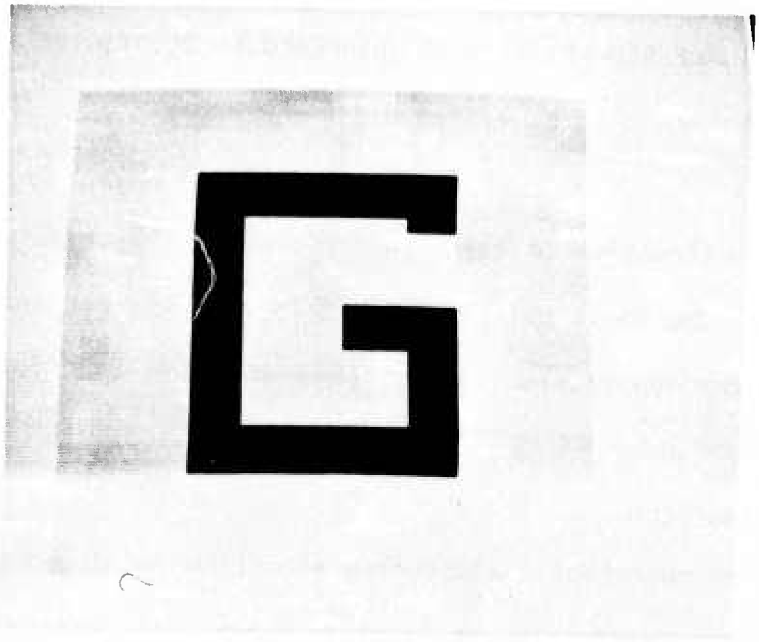


Figure 4 The image of Fig. 3 after the edges have been restored along the horizontal and the vertical directions

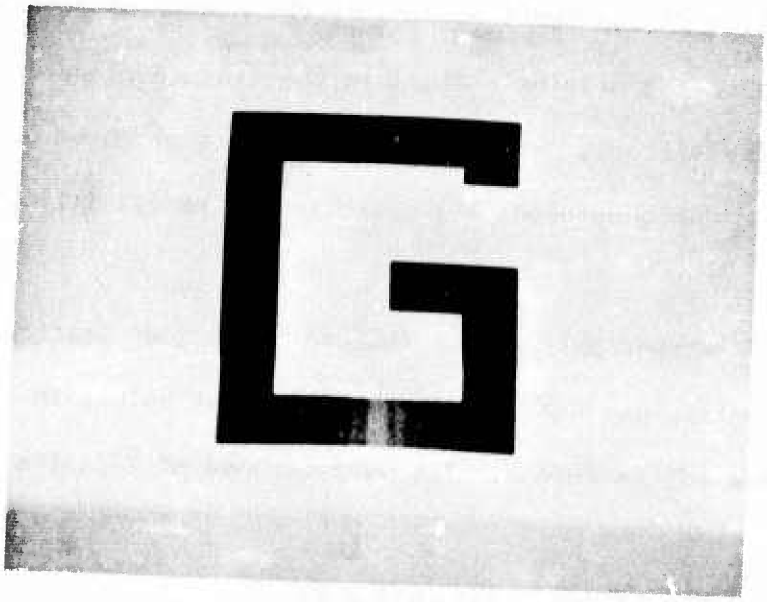


Figure 5 The image of Fig. 4 with grey levels of zero probability mapped to nearest grey levels of nonzero probability

IMAGE SEGMENTATION BY UNSUPERVISED CLUSTERING III

M. Y. Yoo and T. S. Huang

1. Introduction

This is a continuation of the work reported in the last two interim reports. The basic philosophy of this work was presented in detail before and only the new clustering algorithm which was used for this work will be discussed.

2. Clustering Algorithm

We use the non-parametric clustering algorithm developed by Fukunaga et. al. [1]. For large sample sizes this algorithm becomes basically the same as the valley seeking algorithm [2].

The only difference in our implementation is that the conditional probability density function (this is not an actually normalized probability function; the value of the function denotes the number of points which have the same features in the original image) in the feature plane is available rather than sampled vactors. The algorithm consists of three major steps: i) determination of neighborhood; ii) searching of roots; iii) assignment of parent.

Let $\overline{X}$ be the integer pair in the feature plane (our features are quantized to integer levels), and $D(\overline{X})$ denote the number of points in the original picture which have the feature $\overline{X}$. The neighborhood of $\overline{X}$ is the set defined as

$$n_\theta(\overline{X}) = \{ \overline{Y} \mid d(\overline{X},\overline{Y}) \leq \theta, \overline{X} \neq \overline{Y} \}$$

where θ is a preset parameter and d means Euclidean distance.

Then the total number of the original image points which has the feature in the neighborhood $n_\theta(\overline{X})$ is

$$N_\theta(\overline{X}) = \sum_{\overline{Y}\varepsilon n_\theta(\overline{X})} D(\overline{Y})$$

We define the factor $G_\theta\ (\overline{X},\overline{Y})$ as

$$G_\theta\ (\overline{X},\overline{Y}) = \frac{N_\theta(\overline{Y}) - N_\theta(\overline{X})}{d(\overline{X},\overline{Y})}$$

Actually $G_\theta(\overline{X},\overline{Y})$ is the discrete version of the directional derivative. Then the algorithm is follows. If $n_\theta(\overline{X})$ is empty, we call $\overline{X}$ root, and if the neighborhood of $\overline{X}$ is not empty, we calculate $S_\theta(\overline{X},\overline{Y})$ defined as

$$S_\theta(\overline{X},\overline{Y}) = \underset{Y\varepsilon n_\theta(\overline{X})}{\mathrm{MAX}}\ G_\theta(\overline{X},\overline{Y})$$

If $S_\theta(\overline{X},\overline{Y})$ is negative, $\overline{X}$ is also called root and if $S_\theta(\overline{X},\overline{Y})$ is positive we call $\overline{Y}$ the parent of $\overline{X}$. $S_\theta(\overline{X},\overline{Y}) = 0$ case is a little bit complex and we recommend the original paper for details.

Therefore, every point in the feature plane are either classified as roots or given the parent relationship. (roots do not have parents.) Every point except roots are merged into one of the searched roots by parent relationship and the number of roots determined that of clusters.

3. Experimental Results

The number of clusters is very much dependent upon the size of the parameter θ which is used when we define the neighborhood. Smaller θ gives too many artificial clusters and larger θ classifies the whole data as a cluster. The appropriate size of θ for this work was 3. For large size data there is a serious storage problem, and some preprocess for "throwing away" of irrelevant data is highly necessary. We shall present some experimental results in a later report.

References

[1] W. G. Koontz, P. M. Narendra, and K. Fukunaga, "A graph theoretic approach to nonparametric cluster analysis." (to be published)

[2] W. G. Koontz and K. Fukunaga, "A nonparametric valley-seeking technique for cluster analysis," IEEE Trans. on Computers, vol. C-21, No. 2, pp. 171-178, Feb., 1972.

TEXTURE BOUNDARY DETECTION

O. R. Mitchell and W. K. Chan

In an attempt to develop efficient image segmentation algorithms, we are concentrating on texture boundary detection. Most edge detection algorithms do not use texture information at all. Our initial algorithms use texture information exclusively. The combination of texture and gradient-type boundary detectors will be done following our development of the texture edge detection.

The present procedure uses the Max-Min method of texture analysis described elsewhere in this report. However, instead of using the method for a texture classification, a max-min feature computation is performed in two opposite directions from a point in the picture. The resulting features are compared. If the features in opposite directions differ significantly, we mark the center point as a candidate for a texture edge. This calculation is performed iteratively as we move the center point across the picture. Peaks in feature differences indicate texture edges as shown in Fig. 1. More resolution to the edges is found by repeating the scan in directions perpendicular to suspected edges.

Future improvement to be added are the automated selection of scan length, L, based on a regional autocorrelation and the syntactic joining of loosely connected edge points to define closed boundaries.

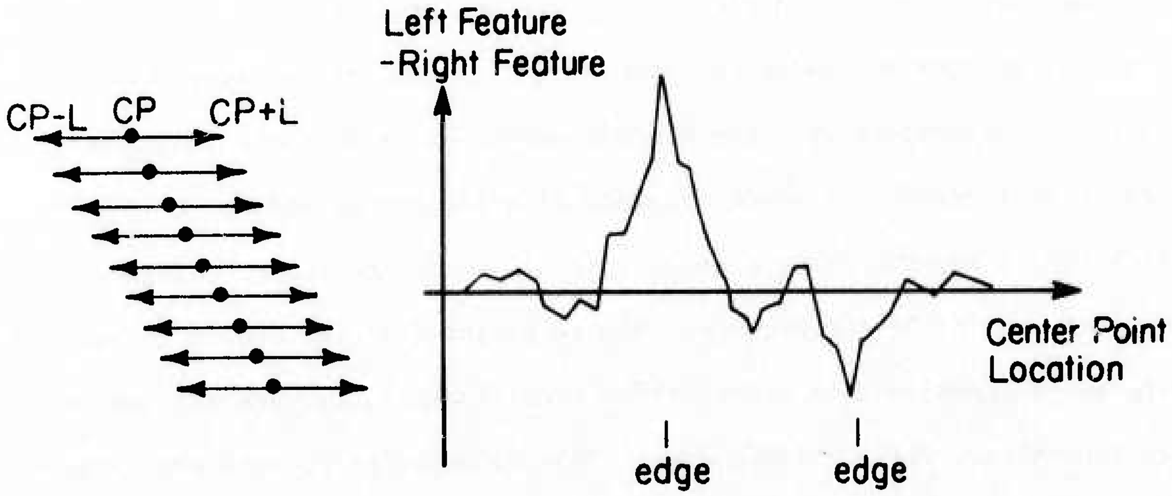


Fig 1. Typical Texture Scan and Resulting Edge Detection

MAX-MIN MEASURE FOR IMAGE TEXTURE ANALYSIS

O. R. Mitchell and W. A. Boyne

1. BASIC MAX-MIN MEASURE

This project is a continuation of that described in the May 1-July 31, 1975 interim report [1]. This texture measure is based on the intuition that important texture information is contained in the relative frequency of local extremes in intensity. The grey levels values along a one-dimensional scan direction are first sent through a smoothing process which eliminates reversals of small amplitude, thereby retaining only the principal extrema.

The smoothing algorithm is described as follows: let x_k be the grey level of the kth point along the scan line and let y_k be the "smoothed" value. Let T be the value of a preassigned threshold parameter. Let us start with $y_1=x_1$ and proceed according to the algorithm shown below:

IF	THEN
$y_k < x_{k+1} - \frac{T}{2}$	$y_{k+1} = x_{k+1} - \frac{T}{2}$
$x_{k+1} - \frac{T}{2} < y_k < x_{k+1} + \frac{T}{2}$	$y_{k+1} = y_k$
$x_{k+1} + \frac{T}{2} < y_k$	$y_{k+1} = x_{k+1} + \frac{T}{2}$

Figure 1 illustrates the way in which the smoothing process eliminates reversals of small amplitude. Implementation of the extrema detecting algorithm is shown in the flow chart in Fig. 2. By repeating this process for several threshold settings, a group of extrema counts can be obtained to characterize the texture. An example is shown in Fig. 3 using three different threshold levels. The set of numbers (6,10,14) would characterize this "texture".

To make the features invariant to multiplicative (gain) changes the logarithm of the data is used. Since

$$\log a - \log b = \log k_a - \log k_b ,$$

the extrema count in Fig. 3 is unchanged following a gain change.

II. CLASSIFICATION RESULTS

The texture data used has been described previously [1]. Forty-nine 64x64 samples of each of eight textures were used. The ratios of the number of extrema at each selected threshold to that at other thresholds were used as features. The threshold settings were chosen empirically to be 130, 110, 90, 70, 50, 30, and 10. (The data was already in log form, ranging from 0 to 255.) Training set data showing the feature values for a few of the texture samples is shown in Table 1.

The 49 samples of each texture were divided into 36 training samples and 13 test samples. Then six features were calculated for each sample and two classification techniques were used: (1) a simple normalized Euclidean distance measure with all features weighted equally, and (2) a three nearest neighbor decision rule. Each texture was classified as a point in 6-dimensional feature space. For the first measure, a mean and standard deviation in each dimension were calculated for each texture from the training samples and the distance for an unknown sample from the test set was measured from the mean in standard deviation units. The results are shown in Table 2. Also included in the table are results using another technique discussed in the next section.

The classification matrix for the method using the 3-nearest neighbors decision rule is shown in Table 3. The most common confusions using the max-min method occur among wood, fur, and water, and between paper and cork. It seems that the distance measure created by the algorithm is similar to that used by humans in grouping textures.

III. COMPARISON WITH SPATIAL-DEPENDENCE TECHNIQUE

The most common texture classifIcation techniques use statistical measures based on spatial dependence probabilities. In order to make the techniques comparable, we limited the technique described by Haralick et.al. [2] to 6 features and to one dimension. We used 64 grey levels in the spatial dependence matrix. The best results we could obtain in using this technique was 83% accurate on training samples and 66% accurate on test samples as shown in Table 2. The classification matrix for the result is shown in Table 4. With 14 features in each of two dimensions the spatial dependence technique accuracy was increased to 94% accurate on training samples and 89% accurate on test samples. However the computation times on the CDC 6500 were indicative of the relative efficiency of the Max-Min Technique. The total feature generation and classification for 49 samples of each of 8 textures required 250 cpu seconds for the 6 feature Max-Min Technique, 845 cpu seconds for the 6 feature spatial dependence technique, and 1690 cpu seconds for the 28 feature spatial dependence technique. However, it should be noted that little effort was made to minimize the running time of any of the above techniques.

IV. DISCUSSION

The max-min appears to be a promising measure of texture characteristics. The method has presently been used in one dimension only. One two dimensional extension would be to measure the max-min features in several directions and use that direction which maximizes some criteria plus the orthogonal direction. This would make the algorithm invariant to texture rotation as well as adding a measure of the rotational symmetry of the textures. The method might also be used to detect texture boundaries by measuring features in two opposite directions from a suspected boundary.

It is fairly easy to incorporate the computing required for the max-min feature extraction in special-purpose hardware. This would make real time texture analysis possible. This is very important for applications such as steel mill output monitoring where a decision must be reached quickly as to whether to let the metal continue cooling or to reprocess it.

Also the quantities measured here (number of extrema vs. threshold) might be called a first order effect. The two curves for cork and paper in Fig. 6 are almost identical and confusions might necessarily be expected in a classification algorithm which uses only slopes of these curves. However, second order measurements which include information as to how the small extrema are interspersed among the large extrema would differentiate between these two textures. This might be the beginning of a hierarchical structure of texture primitives: those that differ in first order measurements and those that differ in second order measurements.

REFERENCES

[1] O. R. Mitchell and W. A. Boyne, "Texture Classification Using Max-Min Techniques," Image Analysis and Modeling, ARPA Interim Report, August 1975.

[2] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," IEEE Trans. Syst., Man, Cybern., Vol. SMC-3, pp. 610-621, Nov. 1973.

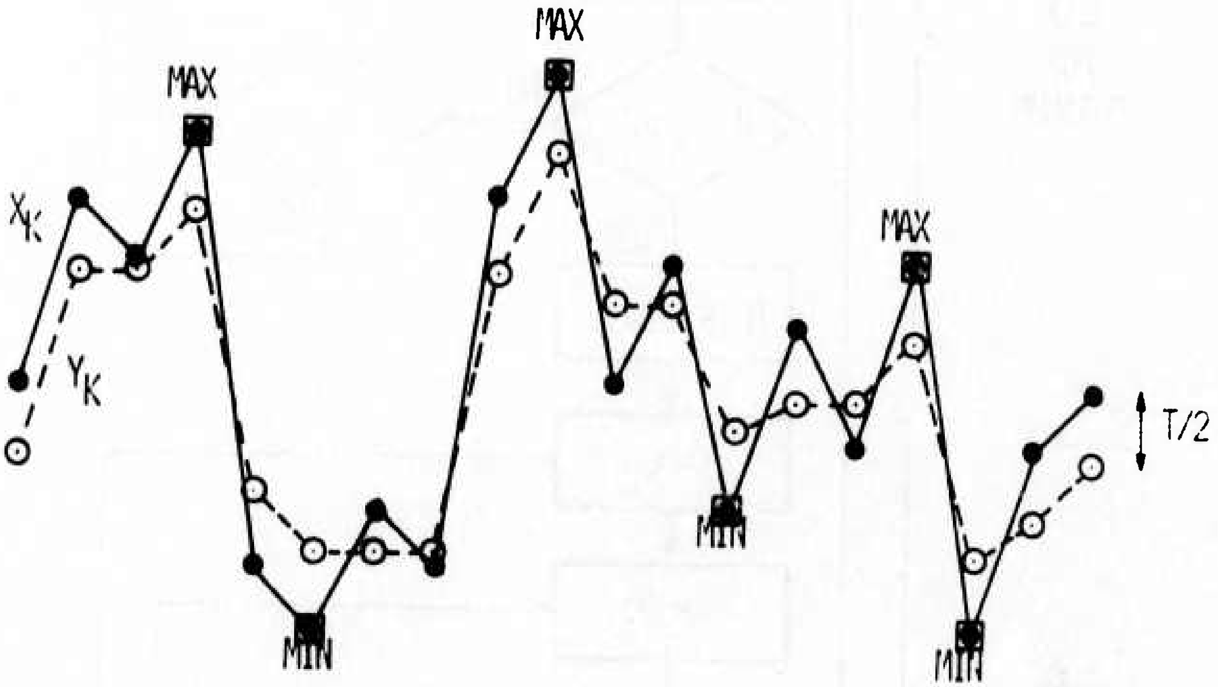


Fig. 1 Backlash smoothing and extrema identification for a threshold distance of T.

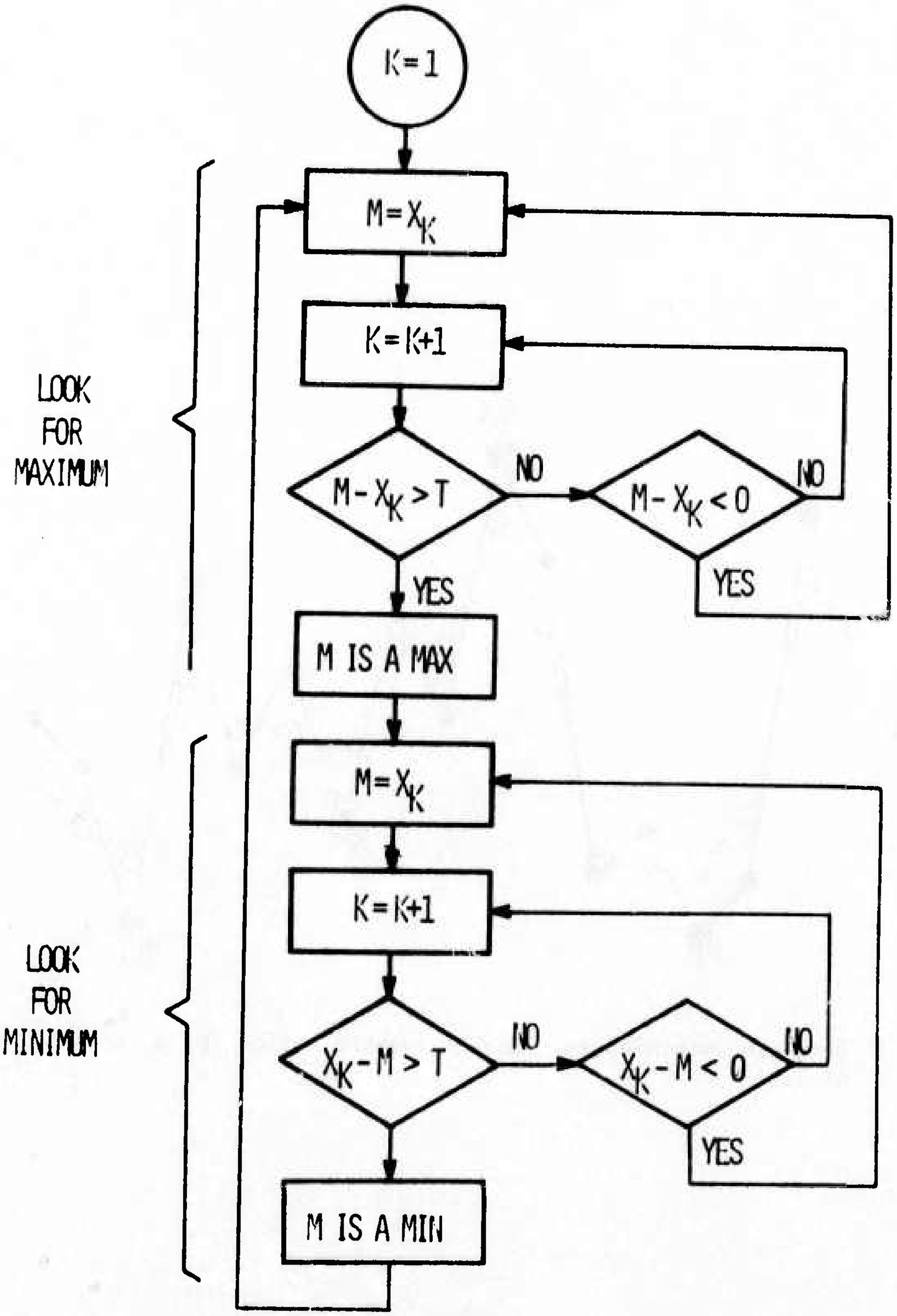


Fig. 2 Flow chart of extrema detecting algorithm. X_K is the grey level of the Kth point. T is the threshold distance.

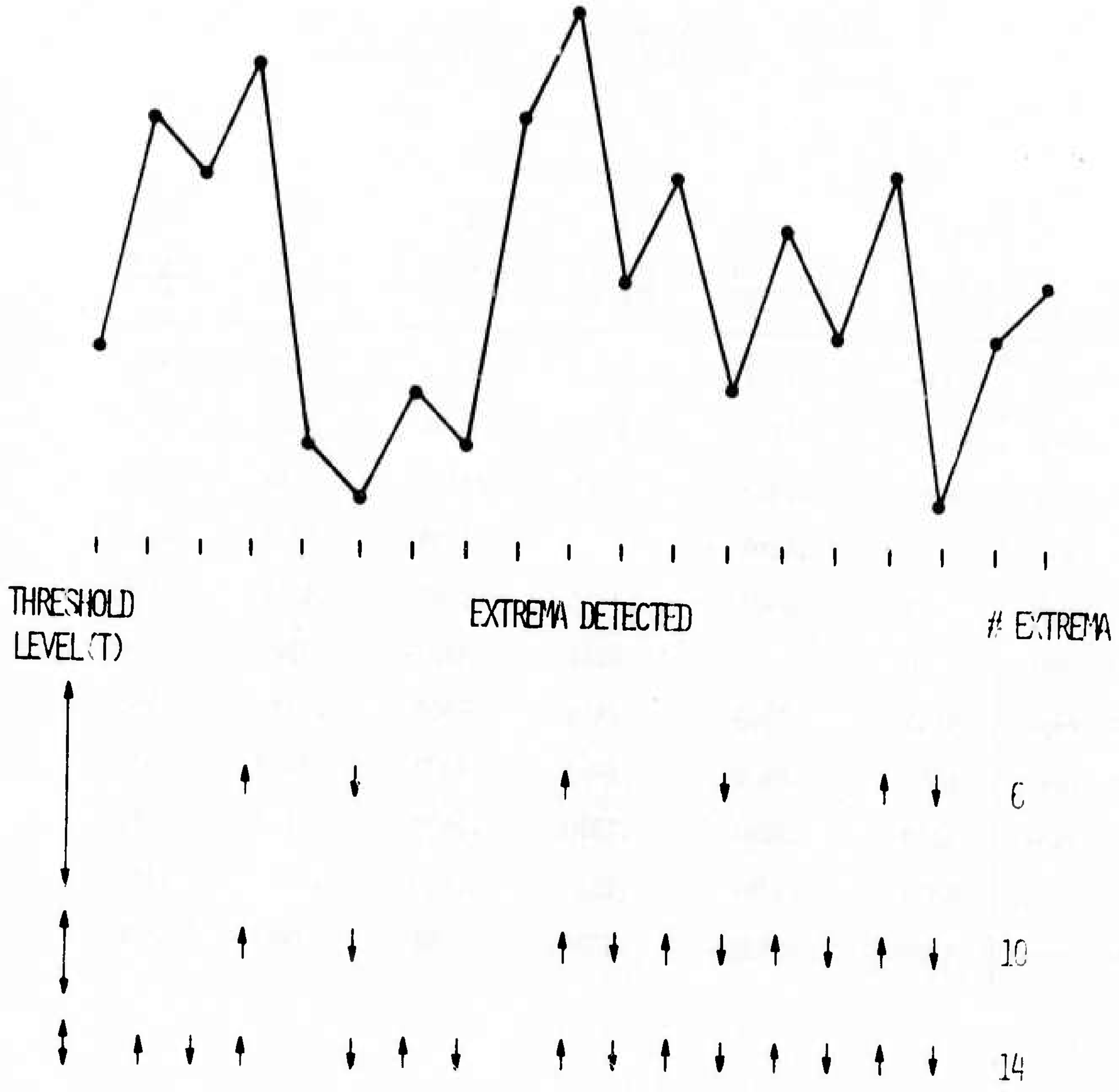


Fig. 3 Extrema detected for three different thresholds.

Table 1 Training Set Data Showing the Feature Values for a Few Texture Samples

	Features					
	1	2	3	4	5	6
Name of Texture	$\frac{\# @ 130}{\# @ 110}$	$\frac{\# @ 110}{\# @ 90}$	$\frac{\# @ 90}{\# @ 70}$	$\frac{\# @ 70}{\# @ 50}$	$\frac{\# @ 50}{\# @ 30}$	$\frac{\# @ 30}{\# @ 10}$
D4 Cork	.7624	.7637	.7338	.5677	.6246	.6793
D4 Cork	.7126	.7767	.6782	.6788	.6227	.6287
D4 Cork	.4957	.6964	.7671	7821	.7467	.5327
D70 Wood	0	.0385	.2653	.3798	.5228	.6520
D70 Wood	0	.0405	.3318	.3940	.5345	.7209
D70 Wood	0	0	.0641	.3805	.3721	.4398
D57 Paper	.5943	.7465	.7513	.8008	.7173	.4301
D57 Paper	.6182	.6875	.7619	.6383	.5557	.5267
D57 Paper	.5319	.6812	.7886	.7479	.7112	.4202
D9 Grass	.6373	.7751	.8328	.7456	.7817	.6636
D9 Grass	.6854	.6893	.7536	.7765	.7447	.6904

Table 2 Classification Results Using 36 Training Training Samples and 13 Test Samples of Each of 8 Texture Patterns

Features Used	% Correct			
	3-Nearest Neighbor Decision Rule		Weighted Distance Decision Rule	
	Training	Test	Training	Test
1. Max-Min Method Extrema Ratios, 6 Features	93.8	83.6	72.9	72.1
2. Max-Min Method Normalized Extrema Ratios, 6 Features	86.8	79.8	77.1	73.1
3. Spatial Dependence Method 6 Features, One Dimension	82.7	66.3	64.9	59.6
4. Spatial Dependence Method 28 Features, Two Dimensions	94.5	89.4	87.2	81.7

Table 3 Classification Matrix for Max-Min Method Using 6 Extrema Ratio Features and 3-Nearest Neighbor Decision Rule

		Assigned Category															
		Training Samples								Test Samples							
Textures		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
D4 Cork	1	32	–	–	–	1	3	–	–	9	–	–	–	–	4	–	–
D70 Wood	2	–	34	–	–	–	–	2	–	–	12	–	–	–	–	1	–
D69 Wood	3	–	–	32	4	–	–	–	–	–	–	10	3	–	–	–	–
D93 Fur	4	–	–	–	35	–	–	–	1	–	–	–	13	–	–	–	–
D29 Sand	5	–	–	–	–	34	2	–	–	–	–	–	–	11	2	–	–
D57 Paper	6	2	–	–	–	2	32	–	–	–	–	–	–	–	13	–	–
D38 Water	7	–	2	–	–	–	–	34	–	–	2	5	–	–	–	6	–
D9 Grass	8	–	–	–	1	–	–	–	35	–	–	–	–	–	–	–	13

Table 4 Classification Matrix for Spatial Dependence Method Using 6 Features and 3-Nearest Neighbor Decision Rule

		Assigned Category															
		Training Samples								Test Samples							
Texture		1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
D4 Cork	1	29	—	—	—	7	—	—	—	7	—	—	—	2	3	1	—
D70 Wood	2	—	25	10	—	—	—	1	—	—	9	3	—	—	—	1	--
D69 Wood	3	—	4	32	—	—	—	—	—	—	4	9	—	—	—	—	—
D93 Fur	4	—	—	1	34	—	—	1	—	—	3	2	8	—	—	—	—
D29 Sand	5	1	—	—	—	30	1	4	—	—	—	—	—	6	1	6	—
D57 Paper	6	1	—	—	—	5	29	1	—	—	—	—	—	4	9	—	—
D38 Water	7	—	—	—	—	2	—	34	—	—	—	—	—	—	—	13	—
D9 Grass	8	1	—	1	—	3	—	—	31	—	—	—	—	4	1	—	8

APPLICATION OF OUTER-PRODUCT EXPANSIONS TO FEATURE EXTRACTION IN PICTURES[*]

K. Fukunaga and G. V. Sherman

A brief description of the two-dimensional version of the subspace method follows. For simplicity square nxn pictures will be assumed. First a representation space is constructed for each of M classes. A typical representation space is characterized by the orthogonal projection operation Q(•)

$$Q(A) = \sum_{i=1}^{m_c} \sum_{j=1}^{m_r} (\phi_i^T A \psi_j) \phi_i \psi_j^T \qquad (1)$$

where A is a random nxn picture, and $m_c \leq n$, $m_r \leq n$. m_c and m_r are the column space and row space dimensionalities respectively and are selected so that about 90% to 95% of each column or row squared norm is contained in the representation space on the average. The $\{\phi_i\}$ and $\{\psi_i\}$ i=1,...,n are arbitrary orthonormal bases. However, two bases in particular have been found to work well. The first is the two-dimensional Karhunen-Loeve transformation [1] in which the $\{\phi_i\}$ are eigenvectors of the sample autocorrelation matrix obtained from all columns of training pictures in a particular class. Similarly the $\{\psi_i\}$ are derived from the sample autocorrelation of picture rows from the same class.

The other basis referred to is obtained by selecting the $\{\phi_i\}$ and $\{\psi_i\}$ as the singular vectors of the singular value decomposition of E{A} [2] where expectation E is restricted to the class in question.

Feature spaces Q′ are obtained from the representation spaces Q by either (2) or (3). Subscripts indicate class labels.

[*] This work was supported in part by NSF Grant GJ-35722.

$$Q_k = Q_k \cap \Big(\bigcap_{\substack{\ell=1 \\ \ell \neq k}}^{M} \overline{Q}_\ell \Big) \tag{2}$$

$$Q_k' = Q_k \cap \overline{\Big(\bigcup_{\substack{\ell=1 \\ \ell \neq k}}^{M} Q_\ell \cap Q_k \Big)} \quad . \tag{3}$$

The intersection of two subspaces is defined to be the subspace contained in both of the original two subspaces [3]. The union of two subspaces is defined to be the subspace consisting of all linear combinations of vectors originating from the original two subspaces. Note that we are dealing with the vector space of nxn arrays.

Our research has shown that (2) and (3) are definitely not equivalent. Part of this phenomenon is due to failure of subspaces to satisfy the distribution law of logic. Namely

$$Q_1 \cap (Q_2 \cup Q_3) \neq (Q_1 \cap Q_2) \cup (Q_1 \cap Q_3) \tag{4}$$

and

$$Q_1 \cup (Q_2 \cap Q_3) \neq (Q_1 \cup Q_2) \cap (Q_1 \cup Q_3) \tag{5}$$

However, further research is needed to study this phenomenon. Algorithms for calculating $Q_1 \cup Q_2$ and $Q_1 \cap Q_2$ exist but will not be discussed for brevity.

Classification is accomplished by minimizing criterion $J^*(\cdot)$.

$$J^*(k_o) = \min_{k} \sum_{i=1}^{m_c} \sum_{j=1}^{m_r} \left[\phi_i^{(k)^T} (A - \acute{E}\{A\}) \, \psi_j^{(k)} \right]^2 \rightarrow A \in \omega_{ko} \tag{6}$$

If class ω_{ko} minimizes $J^*(\cdot)$ over all other classes ω_k then random picture A is classified into class ω_{ko}.

Experiments with this method on the 24x24 Munson handwritten numerals demonstrate a phenomenal feature extraction capability. A 75% reduction in the number of features, from 576 to 144, results in a mere 2.5% decrease in classification accuracy.

Further research is needed to relate the intersection of feature subspaces to intrinsic dimensionality, data dimensionality, number of classes, and classification error. The power of this method lies in its reliance on data structure determined by training samples rather than dubious parametric forms.

Preliminary investigations in these areas are promising. For instance we already found that M subspaces always intersect if the sum of their dimensionalities is strictly greater than (M-1) times the dimensionality of the observacion space. So the problems are solvable.

References

1. K. Fukunaga and G. V.Sherman, "Picture representation and classification by pseudo eigenvectors," 1973 IEEE Systems, Man and Cybernetics Conference, Boston, Mass., November 1973.

2. C. Lanczos, Linear Differential Operators, Van Nostrand, London, 1961.

3. S. Watanabe and N. Pakvasa, "Subspace methods in pattern recognition," International Joint Conference on Pattern Recognition, Washington, D.C., October, 1973.

A BRANCH AND BOUND CLUSTERING ALGORITHM[*]

K. Fukunaga and P.M. Narendra

In clustering, each of N samples is assigned to one of clusters $\omega_1,\dots,\omega_M$, where the number of clusters, M, is assumed to be prespecified. Then, there are M^N possible assignments and they are expressed in a tree form as follows:

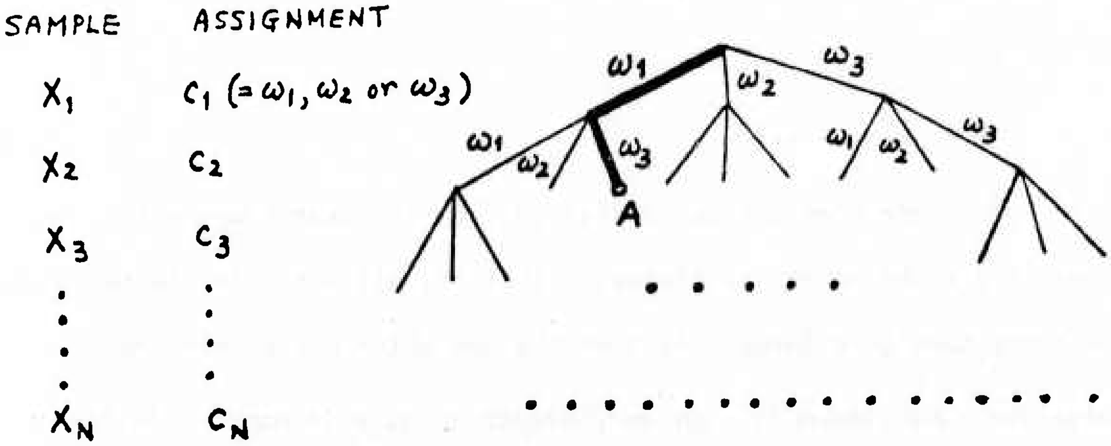


Fig. 1

In order to apply the branch and bound method successfully, we have to come up with a proper criterion to evaluate each node so that the branches under the node can be eliminated from the search.

One of the popular criteria which has been used in clustering frequently is $J = \text{tr}\, S_m^{-1} S_w$ where S_m and S_w are the mixture scatter matrix and the within-class scatter matrix respectively. Since the coordinate can be selected to satisfy $S_m = I$, we may use $J = \text{tr}\, S_w$. This criterion was first used in ISODATA [1], [2]. Thus, our problem is to find the cluster assignments, $c_1,\dots,c_N$ so as to minimize tr S_w.

[*] This work was supported in part by NSF Grant GJ-35722

When the search comes to a node at the k-th level, $c_1,\ldots,c_k$ are already given (for example, $c_1 = \omega_1$ and $c_2 = \omega_3$ for the node A of Fig. 1), but $c_{k+1},\ldots,c_N$ could be anything. Therefore, if we can calculate either $\min_{c_{k+1},\ldots,c_N} J(c_1,\ldots,c_k, c_{k+1},\ldots,c_N)$ or the lower bound of that

$$L(c_1,\ldots,c_k) \le \min_{c_{k+1},\ldots,c_N} J(c_1,\ldots,c_k,c_{k+1},\ldots,c_N) \tag{1}$$

then with the satisfaction of

$$B \le L(c_1,\ldots,c_k) \tag{2}$$

all branches under the node can be eliminated from the search where B is the current lowest J found up to the present. That is, all possible cluster assignments under the node give larger J's than the one which was already found. The tighter the lower bound is, the more branches are eliminated effectively from the search.

Our study revealed that the within-class scatter criterion has a nice property as

$$N\, J(c_1,\ldots,c_N) \ge k\, J(c_1,\ldots,c_k) + (N-k)\, J(c_{k+1},\ldots,c_N) \tag{3}$$

where $J(c_1,\ldots,c_k)$ is calculated from the within-class scatter matrix of $X_1,\ldots,X_k$ with cluster assignment $c_1,\ldots,c_k$, and $J(c_{k+1},\ldots,c_N)$ is for $X_{k+1},\ldots,X_N$ with $c_{k+1},\ldots,c_N$. Let $J^*(c_{k+1},\ldots,c_N)$ be the criterion value for the optimum cluster assignment when only a subset $\{X_{k+1},\ldots,X_N\}$ is considered. Then, $J^*(c_{k+1},\ldots,c_N) \le J(c_{k+1},\ldots,c_N)$, and (3) is bounded from the lower side as

$$N\, J(c_1,\ldots,c_N) \ge k\, J(c_1,\ldots,c_k) + (N-k)\, J^*(c_{k+1},\ldots,c_N) \tag{4}$$

The right-hand side of (4) is independent of the cluster assignments for $X_{k+1},\ldots,X_N$, and provides the lower bound of (1).

There are many possible ways to calculate $J^*(c_{k+1},\dots,c_N)$. One is to start from $J^*(c_N)$ and to calculate $J^*(c_{N-1},c_N)$, $J^*(c_{N-2}, c_{N-1}, c_N),\dots$ successively. The branch and bound method with the node evaluation of (4) gives an efficient way to do this successive extension. When the number of samples becomes large, we may divide the samples into L groups, $\{X_1,\dots,X_{k_1}\}$, $\{X_{k_1+1},\dots,X_{k_2}\},\dots,$ $X_{k_{L-1}+1},\dots,X_{k_L}\}$ $(k_L = N)$. The cluster assignment of each subset is optimized as was mentioned above, keeping the results of the intermediate optimizations, $J^*(c_{k_i})$, $J^*(c_{k_i-1}, c_{k_i}),\dots$. Then, two subsets, for example $\{X_1,\dots,X_{k_1}\}$ and $\{X_{k_1+1},\dots,X_{k_2}\}$, are combined to form a new la larger subset. For $1 \leq k < k_1$, the lower bound is computed by

$$(k_1 + k_2)\, J(c_1,\dots,c_{k_1+k_2}) \geq k\, J(c_1,\dots,c_k) + (k_1 - k)\, J^*(c_{k+1},\dots,c_{k_1})$$
$$+ (k_2 - k_1)\, J^*(c_{k_1+1},\dots,c_{l_2}) \qquad (5)$$

Thus, L subsets are grouped to larger L/2 subsets, and they are again grouped to L/4 subsets and so on.

The procedure to calculate the lower bound mentioned above is only one example among many possibilities. We would like to try many others. Also, we would like to apply this idea to other criteria, particularly to the valley-seeking clustering.

References

1. G. H. Ball and D. J. Hall, "ISODATA, a novel method of data analysis and pattern classification," Stanford Research Inst., Menlo Park, Calif., Tech. Rept., April, 1965.

2. G. H. Ball, "Data analysis in the social sciences: what about the details?" 1965 Fall Joint Computer Conf., AFIPS Proc., Vol. 27, pt. 1, Washington, D.C., Spartan, 1965, pp. 533-559.

IMAGE RESTORATION USING THE PROJECTION METHOD ALGORITHM

T.S. Huang, M. Kaveh and S. Berger

1. Introduction

The projection algorithm has shown promise as an effective method of image restoration. It is well-suited to the treatment of linear spatially-variant degradations. Also, certain types of a priori information that may be available about the original image can be easily incorporated into the restoration process.

The algorithm has been applied to both one and two-dimensional signals in order to evaluate its potential as a restoration technique. The results which are outlined in this report indicate that the algorithm can be an effective method for the restoration of images.

2. The Algorithm

If the degradation of a two-dimensional image can be represented by $g(x,y) = D[f(x,y)] + n(x,y)$, where $f(x,y)$ is the original image, $n(x,y)$ is noise, D is a degrading operator, and $g(x,y)$ is the degraded image, then the purpose of the restoration process is to approximate the original image as closely as possible. The degrading operator is assumed to be linear although space-variant.

If we neglect noise, the discretized version of the degradation is:

$$\begin{aligned} g_1 &= a_{11}f_1+a_{12}f_2+\ldots+a_{1N}f_N \\ g_2 &= a_{21}f_1+\ldots \qquad +a_{2N}f_N \\ &\vdots \\ g_M &= a_{M1}f_1+\ldots \qquad +a_{MN}f_N \end{aligned}$$

where N may not equal M. The projection algorithm is an iterative technique for solving this system of equations.

The solution is obtained by successive projections onto planes in hyperspace. For the case where a unique solution does exist, the algorithm will converge to the point of intersection of the hyperplanes. If the planes do not intersect at a single point, the algorithm will converge to a point which may be a useful approximation in a restoration sense.

The mathematical form of the algorithm is given in vector form. Let $\underline{f} = (f_1,f_2,\ldots,f_N)$ be a point in N space. If the initial guess at a solution is $\underline{f}^{(o)} = f_1^{(o)},f_2^{(o)},\ldots,f_N^{(o)}$, the new solution is the projection of the point $\underline{f}^{(o)}$ onto the hyperplane $g_1 = a_{11}f_1+\ldots+a_{1N}f_N$. So

$$\underline{f}^{(1)} = \underline{f}^{(o)} - \frac{[\underline{f}^{(o)}\cdot\underline{a}_1-g_1]\ \underline{a}_1}{\underline{a}_1\cdot\underline{a}_1}$$

where $a_1 = (a_{11},a_{12},\ldots,a_{1N})$. The next iteration consists of the projection of $\underline{f}^{(1)}$ onto the hyperplane $g_2 = \underline{a}_2 \cdot \underline{f}$ where $\underline{a}_2 = (a_{21},\ldots,a_{2N})$. This process is continued to the Mth plane. This completes one cycle of iteration.

3. Results

The effectiveness of the projection method has been evaluated for several test cases. One test involved a spatially variant degradation in the horizontal direction. The original test image was an "X" in a 128x128 pictule array. The intensity of each picture element was represented by an integer from 0 to 255. The image arrays were stored on magnetic tape in a compacted format. The Gould electrostatic plotter was used to obtain halftone reproductions of the images. The restoration was implemented on a CDC 6500 computer.

The degradation was a smearing which increased its effect toward the edges of the image. The points on the vertical center line were not affected.

By applying the projection algorithm, the image was made increasingly sharper with each iteration. However, the effect of any noise was also increased. So the user would have to decide which iteration yielded the "best" restoration.

The performance of the projection method was also investigated for a one-dimensional signal. The original signal consisted of two unit pulses separated by a small distance. The degraded signal was obtained in the Fourier transform domain by the use of a triangular low-pass filter. The resultant degradation caused the two pulses to be smeared into one wide pulse.

The projection algorithm is capable of utilizing a priori information about the signal. For instance if the signal is known to be positive or be zero in certain regions, the algorithm can include this knowledge in the restoration process. Severla cases were treated with various amounts of a priori information. The algorithm was capable of resolving the image into distinct pulses after a few iterations. In general, the addition of more a priori data improved the performance. The projection method compared favorably with the least-squares inverse filter method, which was also employed.

Two-dimensional images of 128x128 points were also tested. The images were digitized versions of photographs. The results indicate that the algorithm can improve on the degraded image, but the effects of noise dominate the image after a certain number of iterations. Each complete iteration requires about 12 seconds of computation time. The optimum number of iterations is a subjective quantity.

All the above experiments involved image degradations which were simulated on the computer. In our recent work, we applied the projection algorithm to an optically degraded image. Specially, we took an image degraded by camera motion

(Fig. 1(a)), digitized it, and fed it into the computer. Then it is restored by Wiener inverse filtering and by the projection aigorithm. The results are shown in Figs. 1(b) and (c). The projection algorithm gave a much better restored imaqe.

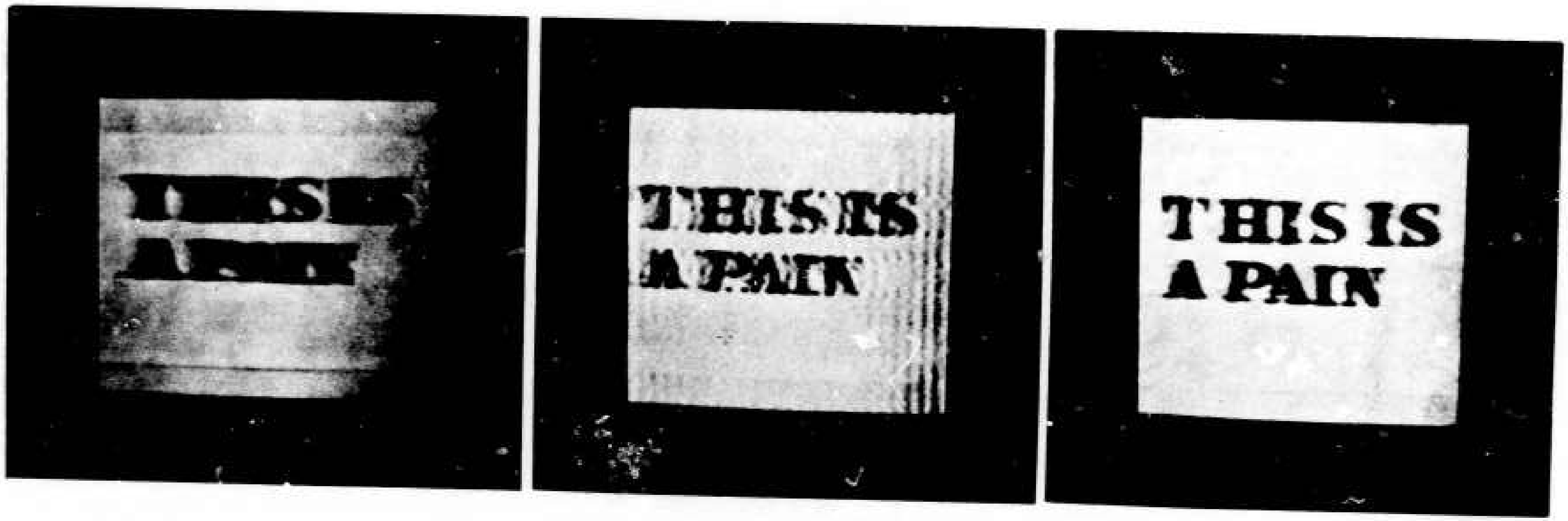
References

[1] K. Tanabe, "Projection Method for Solving a Singular System of Linear Equations and its Applications," Numer. Math. Vol. 17, pp. 203-214, 1971.

[2] T. S. Huang and D. Barker, "Iterative Image Restoration," Summary Report of Research, TR-EE 74-22, School of Electrical Engineering, Purdue University, June 1974.

[3] T. S. Huang, D. A. Barker and S. P. Berger, "Iterative Image Restoration," Applied Optics, Vol. 14, No. 5, May 1975.

[4] T. S. Huang, P. M. Narendra, "Image Restoration by Singular Value Decomposition," Summary Report of Research, TR-EE 75-2, School of Electrical Engineering, Purdue University, November 1974.



(a) (b) (c)

Fig. 1 (a) An image degraded by camera motion. (b) Restored image by Wiener inverse Filtering. (c) Restored image by projection algorithm.

ESTIMATING THE IMPULSE RESPONSE OF A DEGRADING SYSTEM

Brian O'Connor and T. S. Huang

One major problem in image processing is the estimating of the impulse response of a degrading system. Once this is found an inverse filter or other algorithm can be found and applied to the image to eliminate the effects of the degrading system. The estimation problem can be broken down into two cases; in the first, the ideal and degraded images are assumed to be known to some extent; and in the second, only the degraded picture is present. This latter case is called blind deconvolution. Introduced below is a technique which can be applied to either of the above cases, but should find wider use in the latter. This method incorporates image segmentation with a modification of Knox's method [1] for multiframe processing to estimate both the magnitude and phase of the degrading system function.

Let v(x,y) be an ideal image which is degraded by a system with impulse response h(x,y). The observed image is $vc(x,y) = v*h$ which can be segmented into many rectangular strips, $vc_i(x,y)$, $i = 1,\ldots,N$. If the extent of h(x,y) is small compared to the duration of the segments $vc_i(x,y)$, then $vc_i(x,y) \simeq v_i*h$. This implies $VC_i(f_1,f_2) = V_i(f_1,f_2)\cdot H(f_1,f_2)$; furthermore, $|VC_i(f_1,f_2)| = |V_i(f_1,f_2)|\cdot|H(f_1,f_2)|$ and $\angle VC_i(f_1,f_2) = \angle V_i(f_1,f_2) + \angle H(f_1,f_2)$. To estimate $|H(f_1\ f_2)|$ the square magnitudes of the N segments are averaged, so

$$|H(f_1,f_2)| \approx \sqrt{\frac{1}{N}\Sigma|VC_1(f_1,f_2)^2/\frac{1}{N}\Sigma|V(f_1,f_2)|^2}$$

In some applications the light distribution of the N regions of the original image vary sufficiently fast and are sufficiently different from one another so that $\frac{1}{N}\Sigma|V(f_1,f_2)|^2$ is approximately a constant. This implies that the approximation of the magnitude is independent of original scene.

In order to estimate the phase of the system function an average autocorrelation is performed.

$$\Sigma \; VC_i(f_1,f_2) \cdot \overline{VC}_i(f_1+\Delta f_1, f_2+\Delta f_2)$$

$$= \; [\Sigma \; V_i(f_1,f_2) \cdot \overline{V}_i(f_1+\Delta f_1, f_2+\Delta f_2)] \; H(f_1,f_2) \cdot \overline{H}(f_1+\Delta f_1, f_2+\Delta f_2)$$

After some manipulations we find

$$\frac{[\Sigma VC_i(f_1,f_2) \cdot \overline{VC}_i(f_1+\Delta f_1, f_2+\Delta f_2)] \, |\Sigma V_i(f_1,f_2) \cdot \overline{V}_i(f_1+\Delta f_1, f_2+\Delta f_2)|}{|\Sigma VC_i(f_1,f_2) \cdot \overline{VC}_i(f_1,\Delta f_1, f_2+\Delta f_2)| \, [\Sigma V_i(f_1,f_2) \cdot \overline{V}_i(f_1+\Delta f_1, f_2+\Delta f_2)]}$$

$$= \frac{H(f_1,f_2) \cdot \overline{H}(f_1+\Delta f_1, f_2+\Delta f_2)}{|H(f_1,f_2) \cdot \overline{H}(f_1+\Delta f_1, f_2+\Delta f_2)|} = e^{j(\theta(f_1,f_2) - \theta(f_1+\Delta f_1, f_2+\Delta f_2)}$$

that the imaginary part of the complex logarithm gives an estimate of the phase in terms of phase differences. In the above manner the phase differences between every pair of adjacent points can be measured over the whole transform plane.

Programs are being written to simulate the above technique and should give better results than procedures which estimate only the magnitude of the degrading system function. A problem could arise in the phase estimation since it is determined by adding phase differences together, which means that the phase errors add, thus producing the variance of the final error to increase linearly.

References

[1] Knox, K.T. and Thompson, B.J., "Recovery of Images from Atmospherically Degraded Short-exposure Photographs," Astrophysical Journal, vol. 193, No. 1, Oct-Nov 1974, p. L45.

[2] Knox, K. T., "Diffraction-Limited Imaging with Astronomical Telescopes."

MISSILE TRACKING ALGORITHMS

T. S. Huang and G. Y. Tang

This project is motivated by the real-time video tracking problems as the U.S. Army White Sands Missile Range. The missile movie frames were supplied to us by Dr. Alton Gilbert of WSMR.

Our preliminary experimentation has indicated that it is possible in many cases to detect and locate a missile by processing single scan lines. More specifically, we have found that it is possible to determine whether a scan line passes through a missile (and the location of the crossing) by cross-correlating the scan line with a paradign. This opens up the possibility of very fast algorithms which scan selectively instead of full frame. Solid state imagors, such as surface acoustical wave and charged-transfer devices, are particularly suitable for selective scanning. These devices can also be used to do crosscorrelation at extremely high speed.

1) Missile Tracking

We describe an example of missile tracking using single scan line cross-correlations. We used 10 evenly spaced scan lines over the missile frame shown in Fig. 1, and crosscorrelate each scan line with the paradign shown in Fig. 2. By applying a suitable criterion to the correlation results, we were able to determine which scan lines passed through the missile and the crossing locations. Then, among the scan lines that passed through the missile, we picked out the uppermost and lowermost scan lines; and we scan at a finer spacing than before a few lines above the uppermost and a few lines below the lowermost to locate the noise and the tail end of the missile (again by single scan line correlations). The result is shown in Figs. 3 and 4.

2) Initial Acquisition of the Missile

In the WSMR application, we know beforehand where the missile will be launched. We can aim our imagor at the launch site and scan repeatedly (and doing crosscorrelation) at high speed one or several lines above the missile site. Then, we can detect the missile as soon as it moves through these scan lines, and try to track it from that point on.

Figure 1 Original Picture

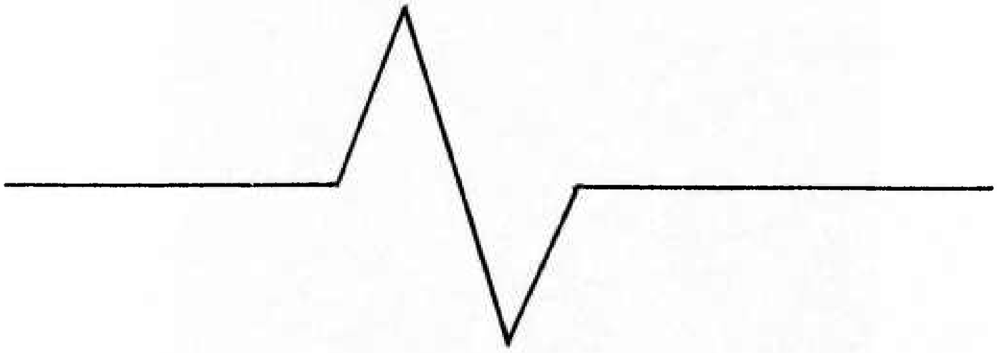


Figure 2 Waveform of Preset Paradigm

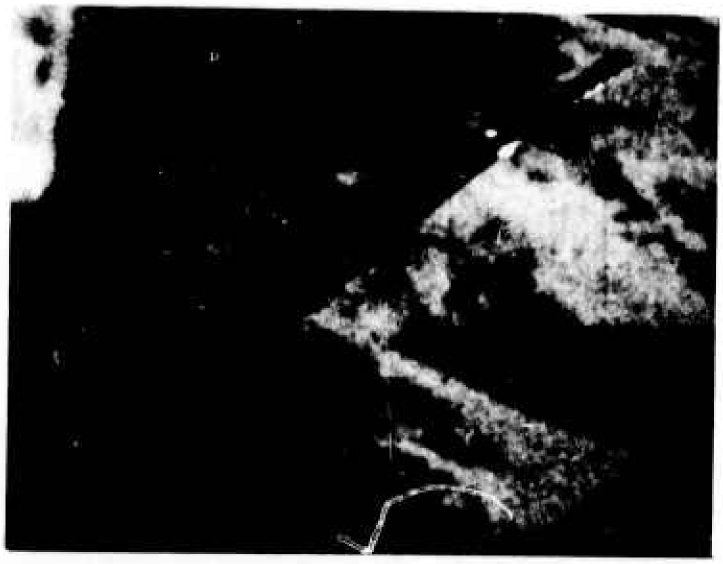


Figure 3 Target Being Taken Out. Dark Area is the Parallelogram Found

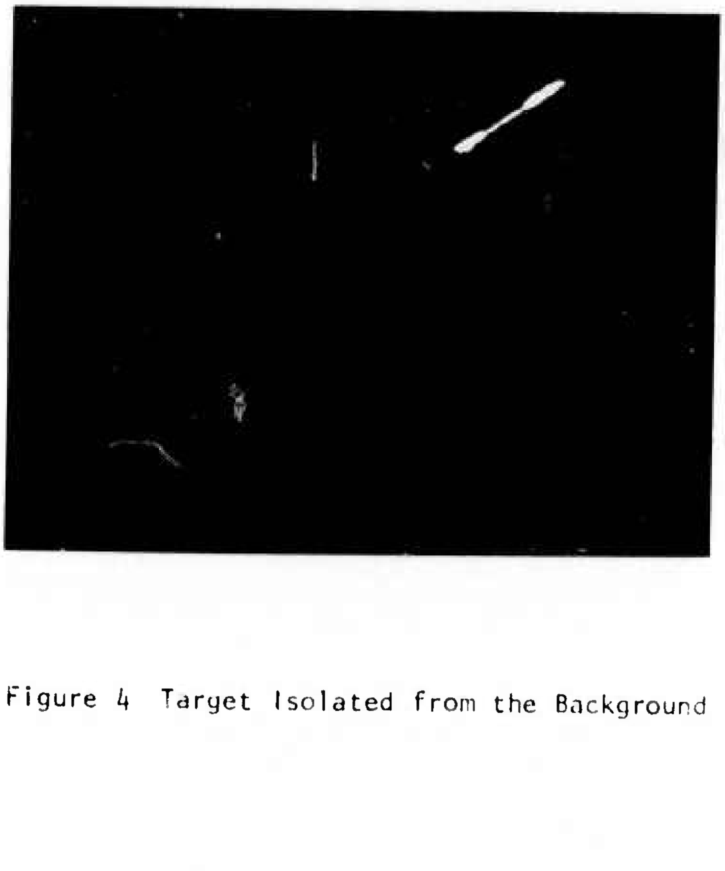


Figure 4 Target Isolated from the Background

INFORMATION EXTRACTION FROM γ-RAYS IMAGES*

G. Y. Tang

Gamma ray cameras have been used by physicians for many years to detect diseases. The grey tone distribution of the picture obtained by the gamma ray camera contains most of the information about the mass density of the patient's tissues or muscles. As a matter of fact, the gamma ray picture is a projection of the mass density of the object illuminated by a radiation source. Physicians make their judgement by comparing an unknown picture with some known diseased patterns. However, since the rich varieties of diseased patterns and the mutual influence between symptoms, the procedure of making judgement is not a straightforward work. Usually it takes two or three years training plus experience. In some cases it is even required a joint judgement of several physicians. The complexity in judgement does not mean that the request from patients should be ignored. A computer-aided-judgement-making algorithm is therefore necessary in order to alleviate the physician's burden and to serve more patients. This is also the ultimate goal of studying gamma ray pictures.

in this report, an attempt has been made to employ pattern recognition techniques to solve the problem of computer-aideded judgement making for gamma ray pictures. As shown in Fig. 1, a pattern recognition system consists of three stages in sequence, i.e., pattern analysis, feature selection and classification. Pattern analysis is to see what is the most informative part which can be obtained. Feature selection is to throw away some unimportant measures and to retain only those which are sufficiently representative so that the computational effort in classification is reduced and that the probability of wrong classification due to the disturbance of unrelevant data is reduced too. The purpose of classification is to assign an attribute for each input pattern.

*This work was supported in part by Consiglio Nazionale delle Ricerche, Italy.

Some pattern analysis experiments whcih have been done are described in the first section of this report. In the second section some possible applications of these measurements for the purpose of combined aid system recognition are discussed briefly.

I. Pattern analysis.

A gamma ray picture is represented by a matrix $\{I(i,j)\}$ $i=1,\ldots,N_x$, $j=1,\ldots,N_y$ where N_x and N_y are the spatial extention of the gamma ray picture in x and y directions. $I(i,j)$ is the (i,j) entry of that matrix. Its value corresponds to the intensity of radiation energy at (i,j) position. In our experiment, $N_x = N_y = 64$, $I(i,j)$ varies between 0 and $2^{16}-1$. Statistically a gamma ray picture is an outcome of a Poisson's process. So $P(I(i,j) = x) =$

$\frac{e^{-m(i,j)} m(i,j)^x}{x!}$, $P(I(i,j) = x)\Delta x$ is the probability of $I(i,j)$ of value x. Poisson's distribution indicates that, for the area of higher mean value, the variance or the noise is larger too. In order to combat the noise disturbance, a smooth technique is required. A two dimensional symmetric low-pass filter is chosen for this purpose. After the noise has been cleared out, we threshold the picture by a preset threshold. The thresholded picture has only two grey tones 1 and 0. 1 is assigned to those picture points with $I(i,j) \geq T$, T is the threshold, 0 is otherwise. The thresholded picture is denoted by $I'(i,j)$. A simple edge detector can be applied to $I'(i,j)$ in order to get contours. The edge detector works as if $|I'(i,j) - I'(i+1,j)| > 0$ then it reports an edge point at (i,j), otherwise it reports no edge points. Fig. 2 shows the contour obtained by setting threshold T=8000 and by using the foregoing edge detector. Notice that the left lobe and the right lobe are not distinguishable. This gives us a hint to try a lower threshold T=4500. Fig. 3 displays the contour thus obtained. Both lobes are not shown. A third trial is a threshold of value 7000. Fig. 4 displays the result. We can see that the two lobes are present, but the left one is rather small. More than that, the deformity which contains most of the symptomatic information is not significant. All these evidences indicate the difficulty to set up a good threshold so that an informative contour may be obtained. In order to combat that, we proposed an approach which should achieve that 1) it guarantees two separate contours for the two lobes 2) it subjects to a better tolerance for the variation of the threshold(This means, for example T=4500 and T=7000 should generate about the same contours). The basic frame for the proposed method is to break up the entire picture into several regions. Then we locate these two regions which enclose the two lobes. New threshold on the original picture for each of these two regions is set up according to the local grey level distribution. Finally, a contour follower is employed to get the coordinates of the points on the contour. The foregoing edge detector is used for the purpose of displaying.

More specifically, the proposed method can be described as:

(1) Picking up a threshold T.

(2) An operation T_h is defined as

$$T_h(i,j) = 1 \text{ if } I(i,j) \geq T$$

$$T_h(i,j) = 0 \text{ if } I(i,j) < T$$

and $I' = T_h(I)$ means $I'(i,j) = T_h(i,j)$. So I' is a matrix. $I(i,j)$ which is the picture matrix has been defined before.

(3) An operation E, which is the edge detector, is defined as:

$$E(i,j) = 1 \text{ if } |I'(i,j) - I'(i+1,j)| > 0 \text{ for all } i \text{ and } j$$

$E(i,j) = 0$ otherwise, where $I' = T_h(I)$ and $I'' = E(I')$ means $I''(i,j) = E(i,j)$ for all i,j.

(4) From I'', we obtain two points TOP, BOTTOM as:

TOP = (i_1,j_1) such that $E(i_1,j_1) = 1$ and j_1 is the maximum of all j such that $E(i\ j) = 1$.

BOTTOM = (i_2,j_2) such that $E(i_2,j_2) = 1$ and j_2 is the minimum of all j such that $E(i,j) = 1$.

(5) A line M is defined as

$$M = \{(i,j) \mid j = \frac{1}{2}(j_1+j_2),\ i = 1,\ldots,N_x\}$$

(6) A line M_1 is defined as

$$M_1 = \{(i,j) | j = \frac{1}{2}(j_2+\frac{1}{2}(j_1+j_2),\ i = 1,\ldots,N_x\}$$

(7) An operation C on each horizontal line, i.e. j = const, is defined as

$$C(j = \text{const}) = \frac{1}{N_j} \sum_{i=1}^{N_x} I'(i,j)\ x i$$

where N_j is the number of 1's on the line j = const on picture I'. This operation locates the centroid on each horizontal line.

(8) For lines lying between M and M_1, we can apply operation C to each of them. Then we obtain a set of M_1-M+1 points (x_1,y_1), (x_2,y_2).....,$(x_{M_1-M+1}, y_{M_1-M+1})$. The first component is obtained by operation C, the second is the one defining the horizontal line.

(9) A line Y = AX + b is found such that $\sum_{i=1}^{M_1-M+1} (y_i-ax_i-b)^2$ is minimum. Y is corresponding to the second element and X is corresponding to the first element in our $I(i,j)$, $I'(i,j)$, $I''(i,j)$ notations.

(10) A line segment $R: Y = ax+b$ where $\frac{1}{2}(j_1+j_2) \le Y \le j_1$ is thus obtained.

(11) The picture is now segmented into four components denoted by A, B, C, D. A is the background. B is what is below the line M but not the background. C is what is to the left of R and what is above M, but not the background. D is the same as C except to the right of R. Fig. 5 shows the segmentation from

Fig. 2. Fig. 6 shows the segmentation from Fig. 3. Fig. 7 shows the segmentation from Fig. 4.

(12) From now on, we are only interested in area C and D. For each of these two areas, we define a histogram from which a new threshold can be obtained. The common maximum and minimum in areas C, and D are located. They are denoted by MAX and MIN respectively. The histogram H_c and H_d is defined as

$$H_c(k) = \# \ \{(i,j) | k \leq \frac{I(i,j)-MIN}{MAX-MIN} \ X ICELL < k+1, -(i,j) EC\} \text{ and}$$

$$H_d(k) = \# \ \{(i,j) | k \leq \frac{I(i,j)-MIN}{MAX-MIN} \ COCELL < k+1, \ (i,j) \ E \ D\}$$

where k = 1,2,...,ICELL, and # denotes the number of elements. The new thresholds T_c and T_d for the area C and D respectively are:

$$T_c = \min \ \{k | \sum_{i=1}^{k} h(i) \geq N_c\}$$

$$T_d = \min \ \{k | \sum_{i=1}^{k} h(i) \geq N_d\}$$

$$N_c = \# \{ (i,j) | (i,j) \ EC\} \ X \ PER \ 1$$

$$N_d = \# \{ (i,j) | (i,j) \ EC\} \ X \ PER \ 2$$

PER 1 and PER 2 are two preset numbers which are 0 < PER 1 < 1 and 0 < PER 2. < 1.

(13) Thresholding and edge detecting same as (2) and (3) are applied to areas C and D with new thresholds T_c and T_d respectively. (PER 1 = PER 2 = 0.4).

Fig. 8 shows the contours obtained from Fig. 2. Fig. 9 shows the contours obtained from Fig. 3. Fig. 10 shows the contours obtained from Fig. 4.

Notice that we can always obtain two separate contours by the three different thresholds, i.e., T = 4500, 7000, 8000. Also, the contours from T = 4500 and T = 7000 are exactly the same and the contour from T = 8000 is very similar to that from T = 4500 and T = 7000. Especially for the good lobes [the right one], they are almost indistinguishable.

(14) A contour follower is designed to replace the edge detector so that it will always report a closed contour no matter what the shape of the contour. The input to the contour follower is a window of size 3x3. The center of the window [0 on Fig. 11] is already known on the contour. One of the eight neighbors of the center is known on the contour too. The function of the contour follower is to find a point which is supposed to be the next point in the eight neighbors around the center. Then the window moves to and centers at that point. Repeating the same procedure, we can obtain the whole contour. The way to locate the next point from the eight neighbors of the center is simply looking at the eight neighbors in counterclockwise sense starting with the neighbor already known on the contour. For example, on Fig. 12, searching starts with 2 and then 3,4,5 until 6 where the next point is found. The contour then reports the coordinate of the next point (i,j) and the direction of moving which is coded into 8 numbers as shown on Fig. 11.

Table 1 lists the two contours obtained by setting T = 8000 and PER 1 = 0.4 and PER 2 = 0.4. The edge detection is replaced by the contour follower. The corresponding direction code is attached too. Fig. 14 shows the scatter distribution of the direction code on Table 1.

(15) The contours obtained from the unfiltered picture by the same method as to obtain Fig. 10 is shown on Fig. 13 where T = 8000, PER 1 = PER 2 = 0.4. Obviously these two contours are more noisy.

Fig. 15 shows the contours by setting T = 8000, PER 1 = 0.6, PER 2 = 0.4 from a filtered picture by the same method as to obtain Fig. 10.

Fig. 16 shows the contour by setting T = 8000, PER 1 = 0.15, PER 2 = 0.4 from a filtered picture by the same method as to obtain Fig. 10.

II. Discussions.

In the previous section, it shows the possibility to obtain two separate contours from a gamma ray picture by the proposed method with which a set of coordinates (x_i,y_i), $i=1,\ldots,N$ is reported. How to use this set of coordinates to make a classifier is another problem. One way to tackle this problem is to perform a transformation on the coordinates so that, after transformation, the new coordinates should be 1) invariant for translation of the object, 2) invariant for the rotation of the object, 3) invariant for the reflection, 4) invariant for the swelling or shrinking of the object. Several methods ha have been studied in order to achieve them. Locating the centroid and using the centroid as the origin of a polar coordinate system is one way. The use of Fourier transformation by treating each point (X_i,Y_i) as a complex number x_i+jy_i, $j = \sqrt{-1}$ is another way. The curvature may be a useful quantity too. However, whatever the transformation is, we will have a new set of coordinates (u_i,v_i), $i=1,\ldots,M$. Generally M is smaller than N. A discriminant function f is defined on $u_i,v_i)$, $i=1,\ldots,M$ such that $f(u_1,v_1,u_2,v_2,\ldots,u_M,v_M) \geq 0$ indicate disease, and $f(u_1,v_1,u_2,v_2,\ldots,u_M,u_M) \geq$ indicates no disease f may be a linear or a nonlinear function. Supervised or non-supervised (learning techniques may be used to get f with a set of training data which we know if any of them is good or diseased.

Accompanying with experiments, further studies of transformations and discriminant functions are necessary to complete the system. The syntactic pattern recognition technique may be adapted too. How acceptable of the syntactic approach depends mainly on experimental results. It is hard to predict. A combination of statistical methods and syntactic methods may bring in a more accurate diagnosis. The employment of different pattern recognition techniques to this particular set of data is the major task in the future.

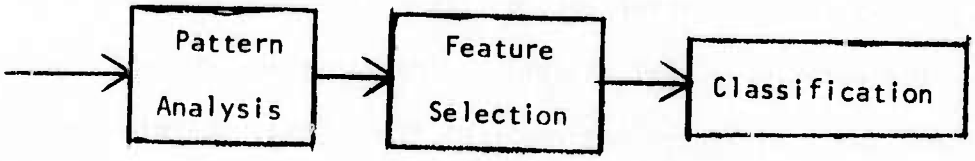


Fig. 1 System for pattern recognition

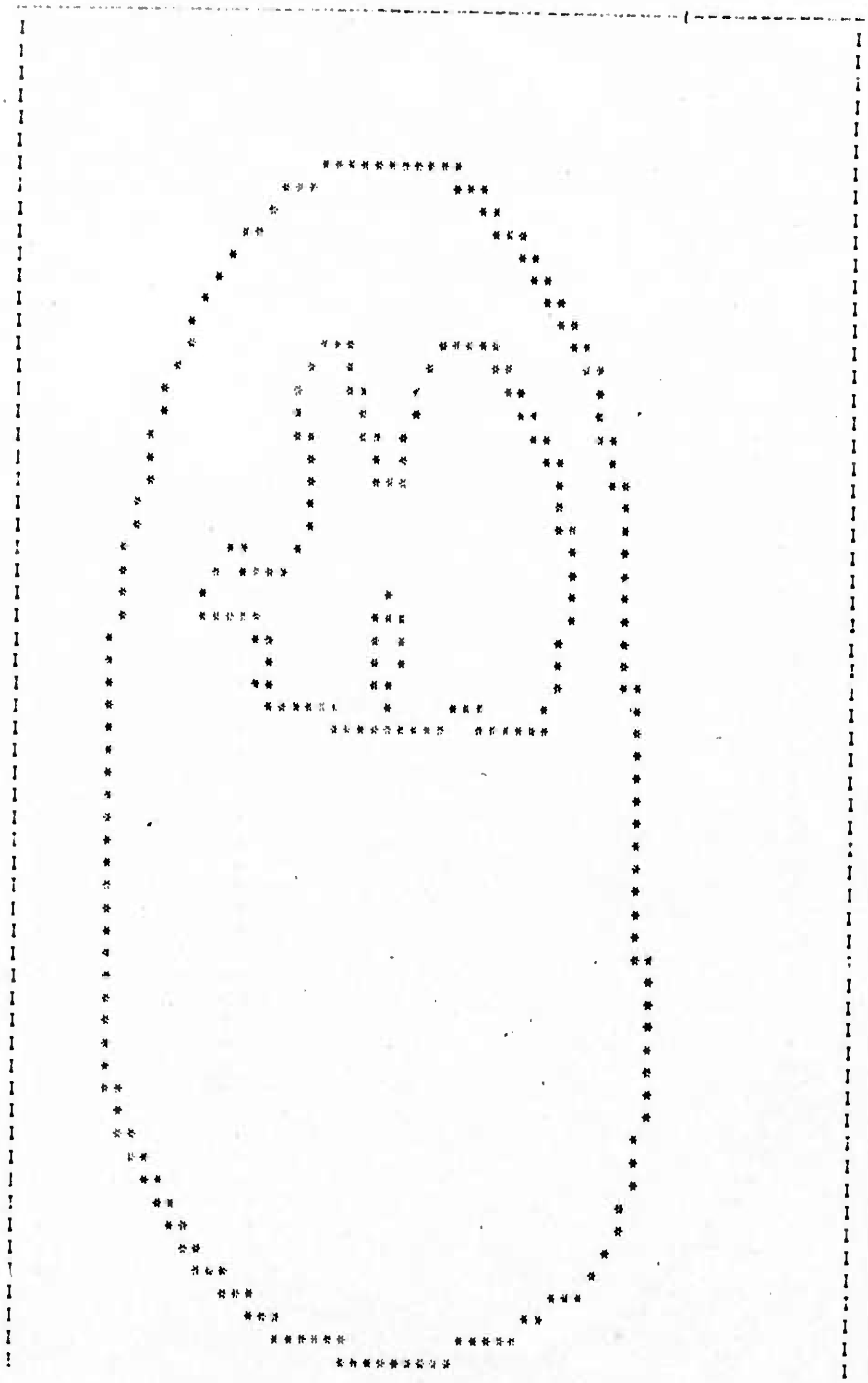


Fig. 2 Edges obtained by setting threshold equal 8000.

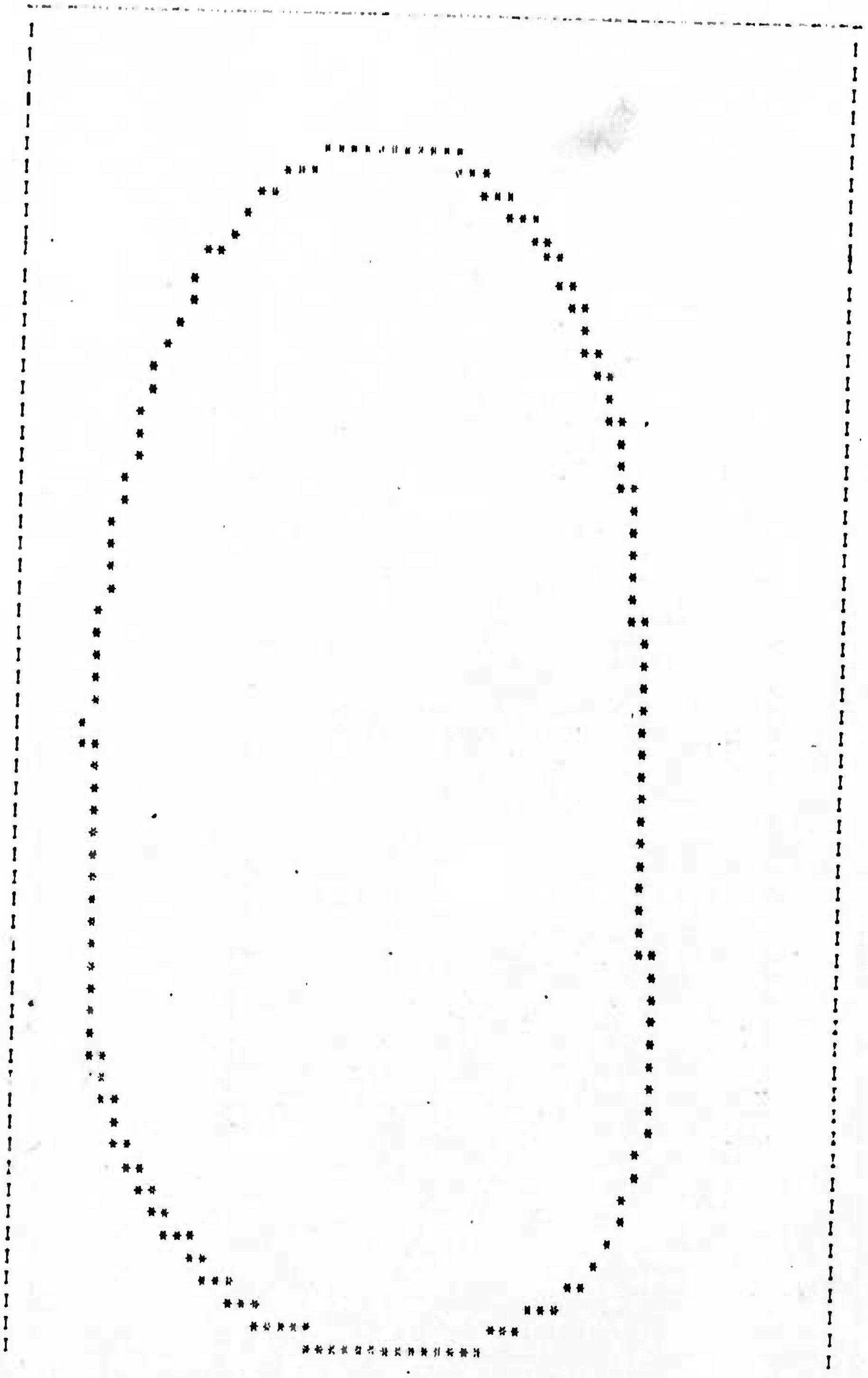


Fig. 3 Edges obtained by setting threshold equal 4500.

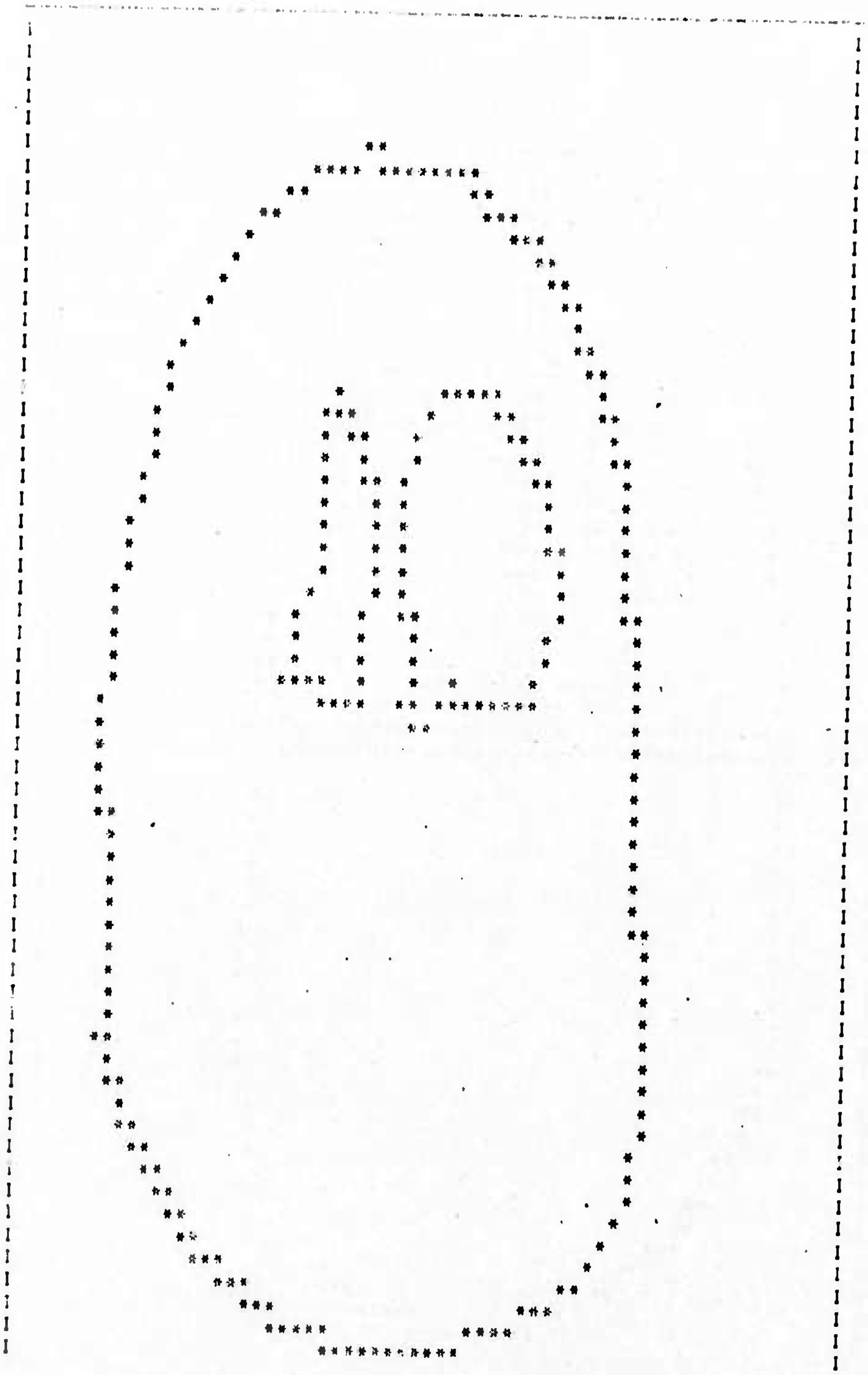


Fig. 4 Edges obtained by setting threshold equal 7000.

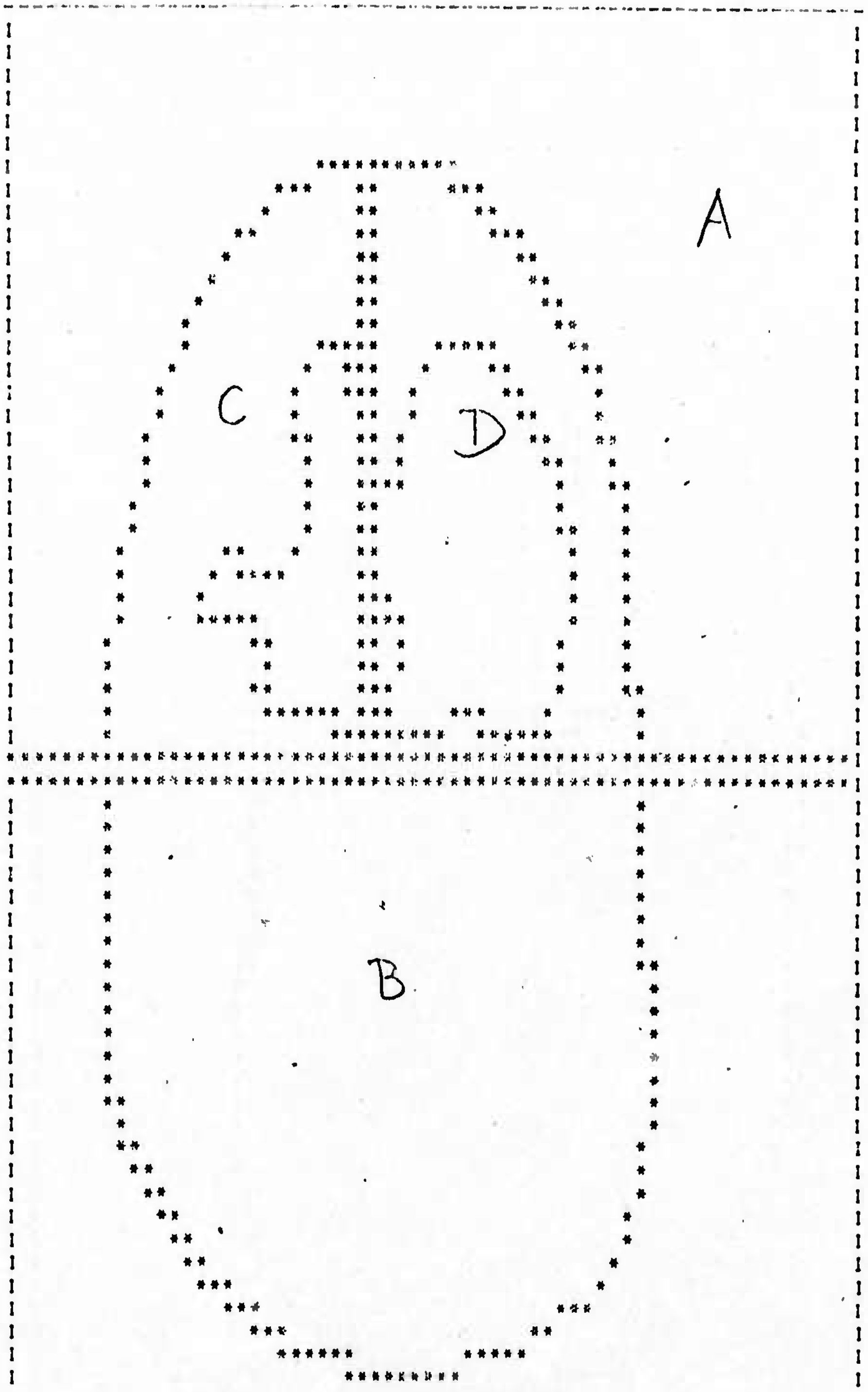


Fig. 5 Four regions obtained from Fig. 2.

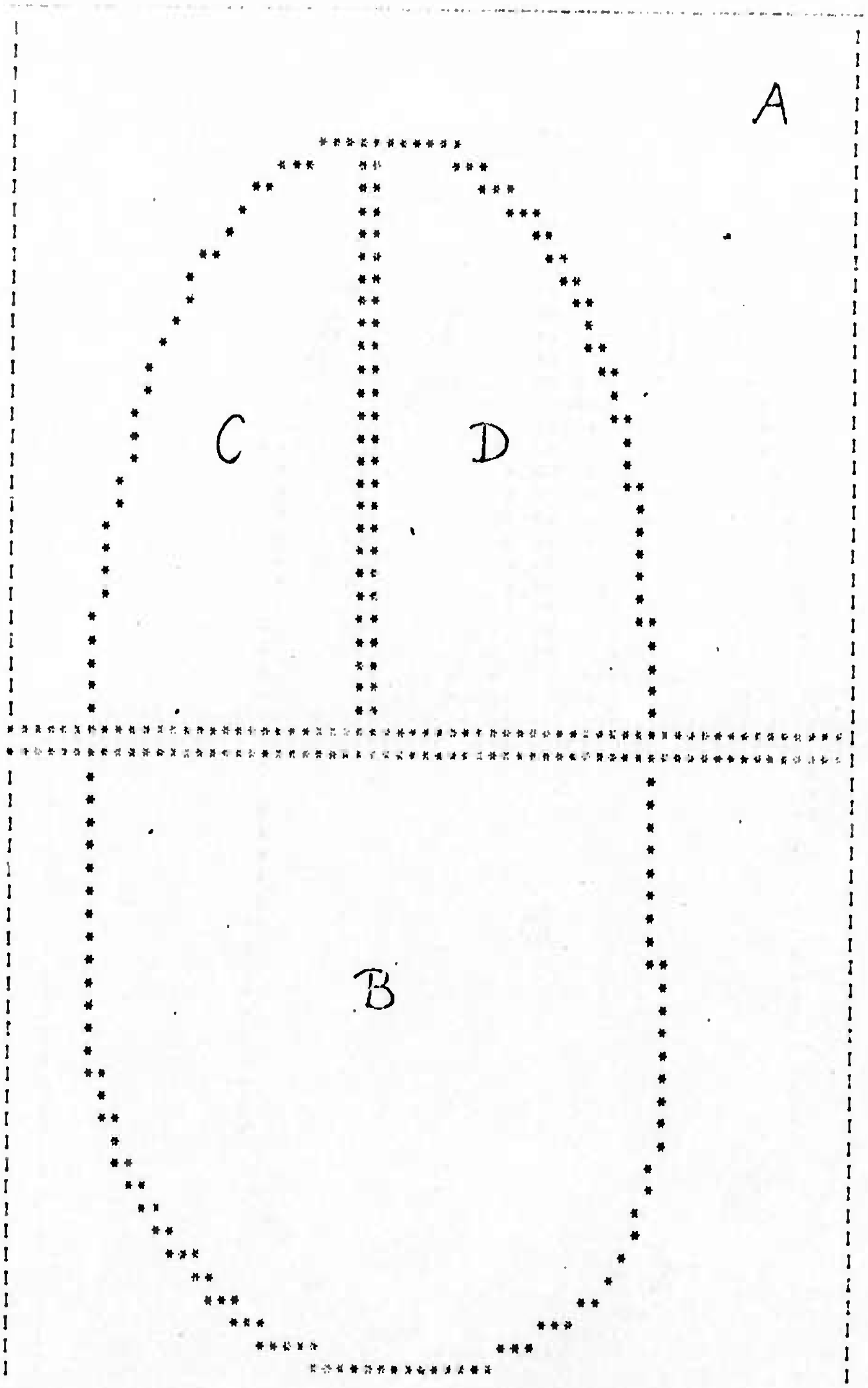


Fig. 6 Four regions obtained from Fig. 3.

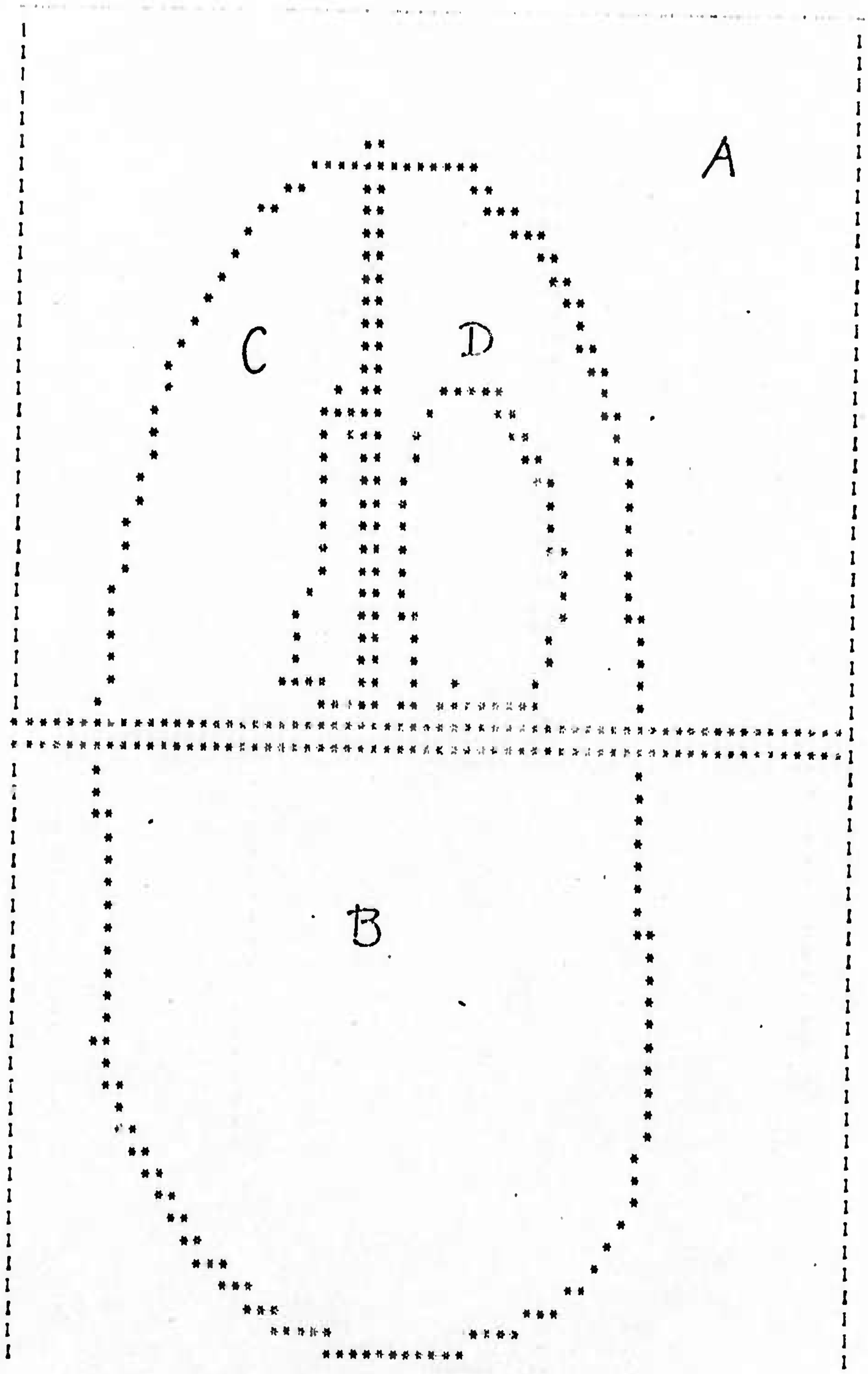


Fig. 7 Four regions obtained from Fig. 4.

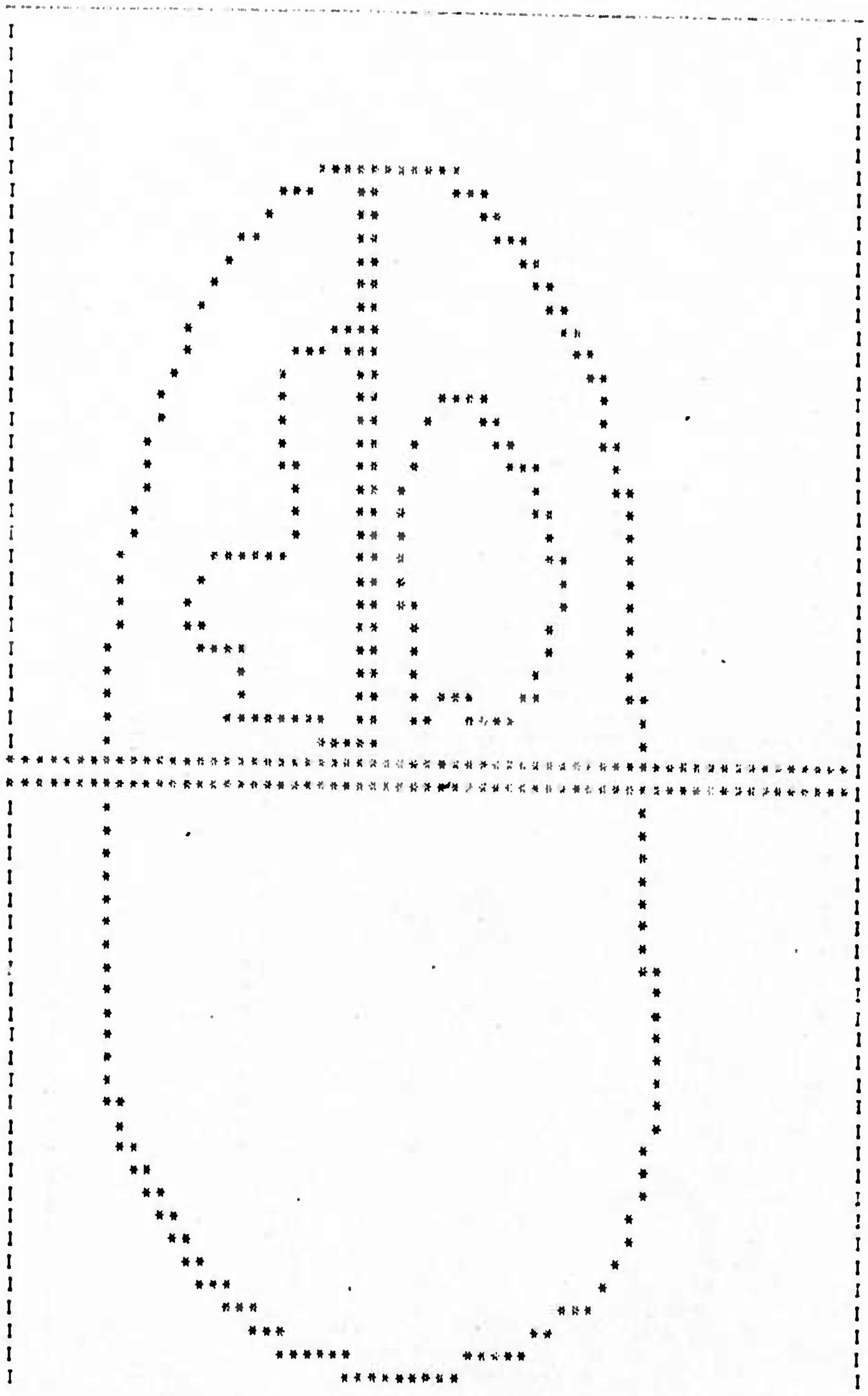


Fig. 8 Contours obtained from Fig. 2.

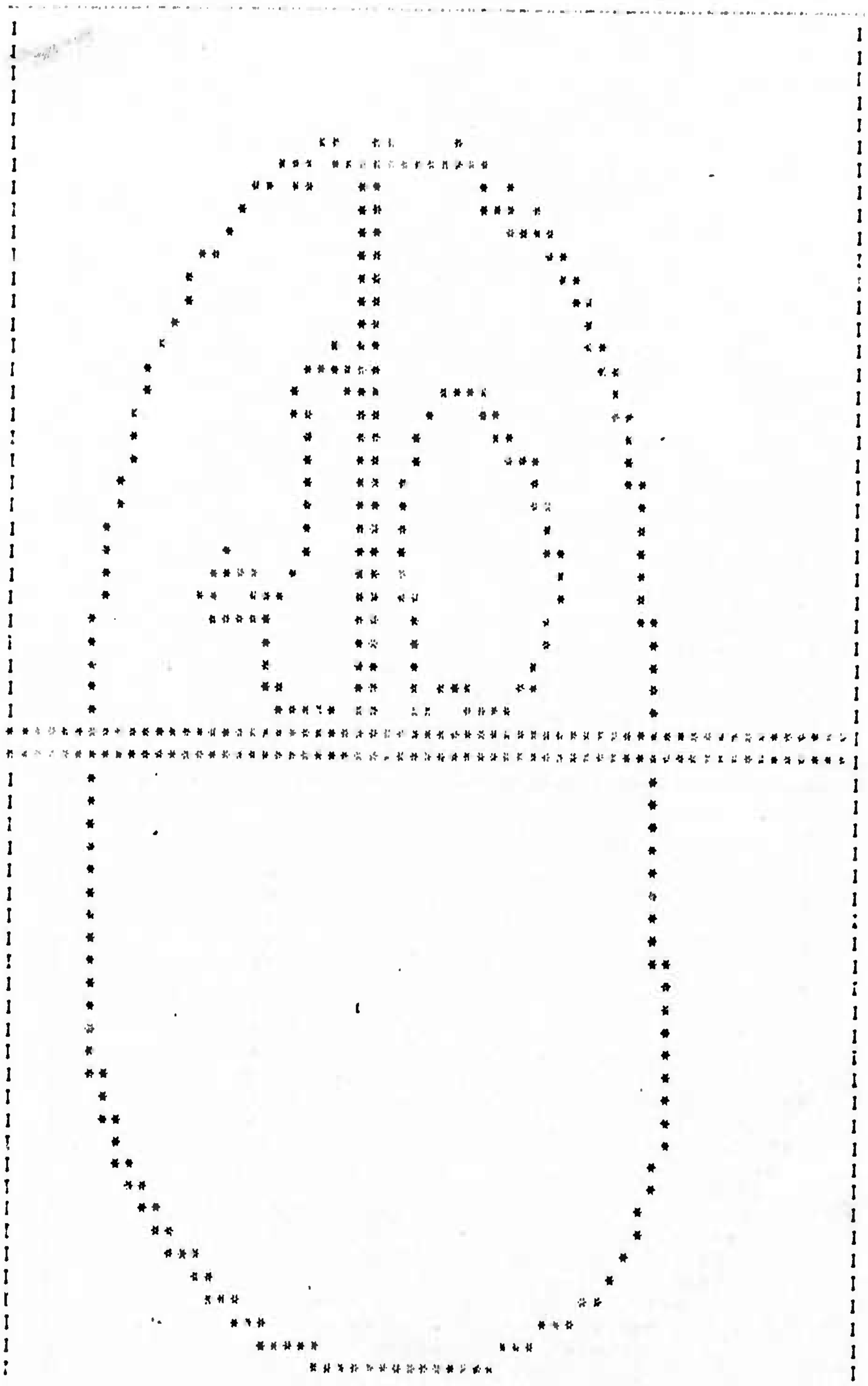


Fig. 9 Contours obtaineu from Fig. 3.

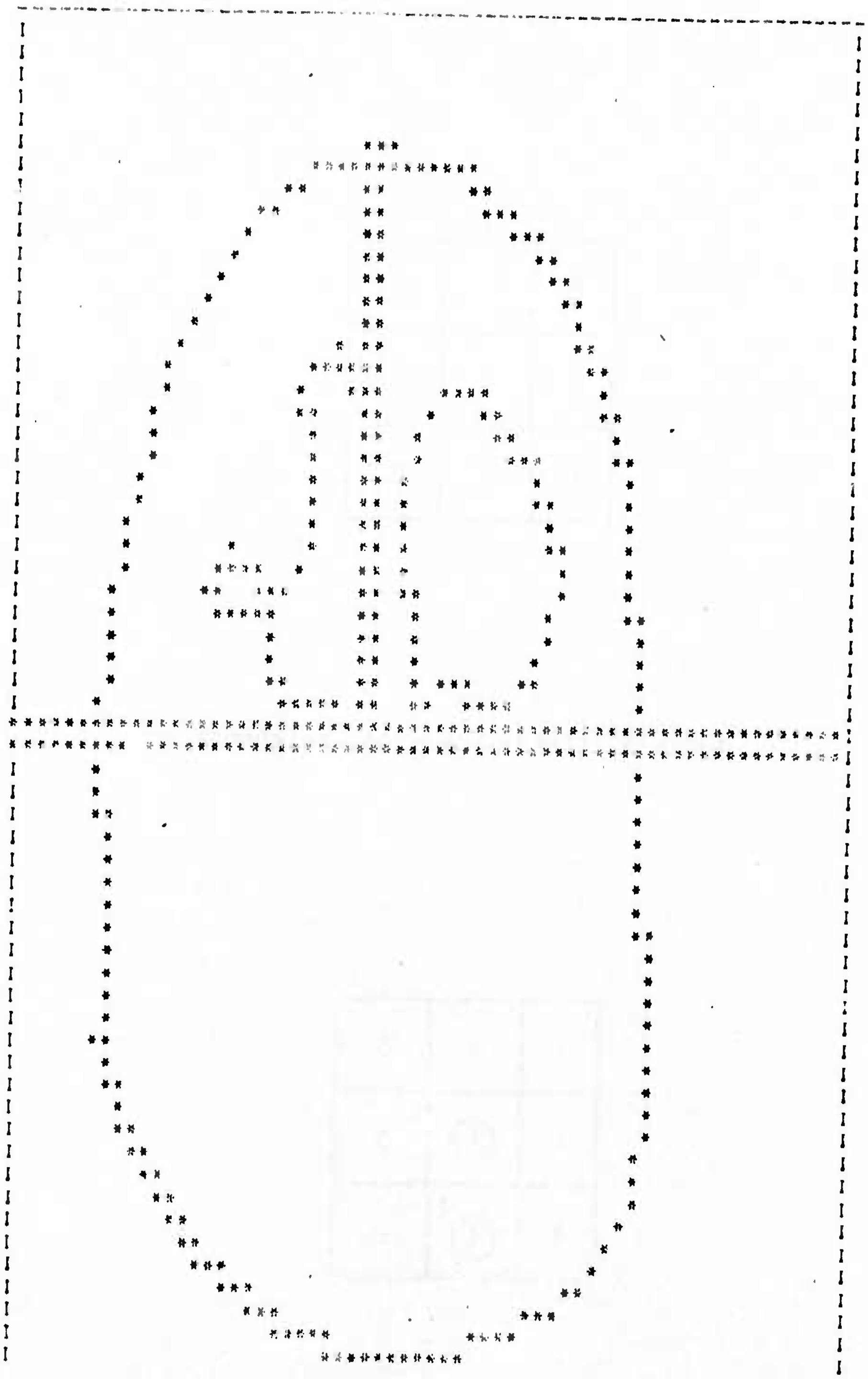


Fig. 10 Contours obtained from Fig. 4.

6	5	4
7	0	3
8	1	2

Fig. 11 Labels for eight neighbors.

1	0	0
1	①	0
1	①*	0

Fig. 12 A contour follower.

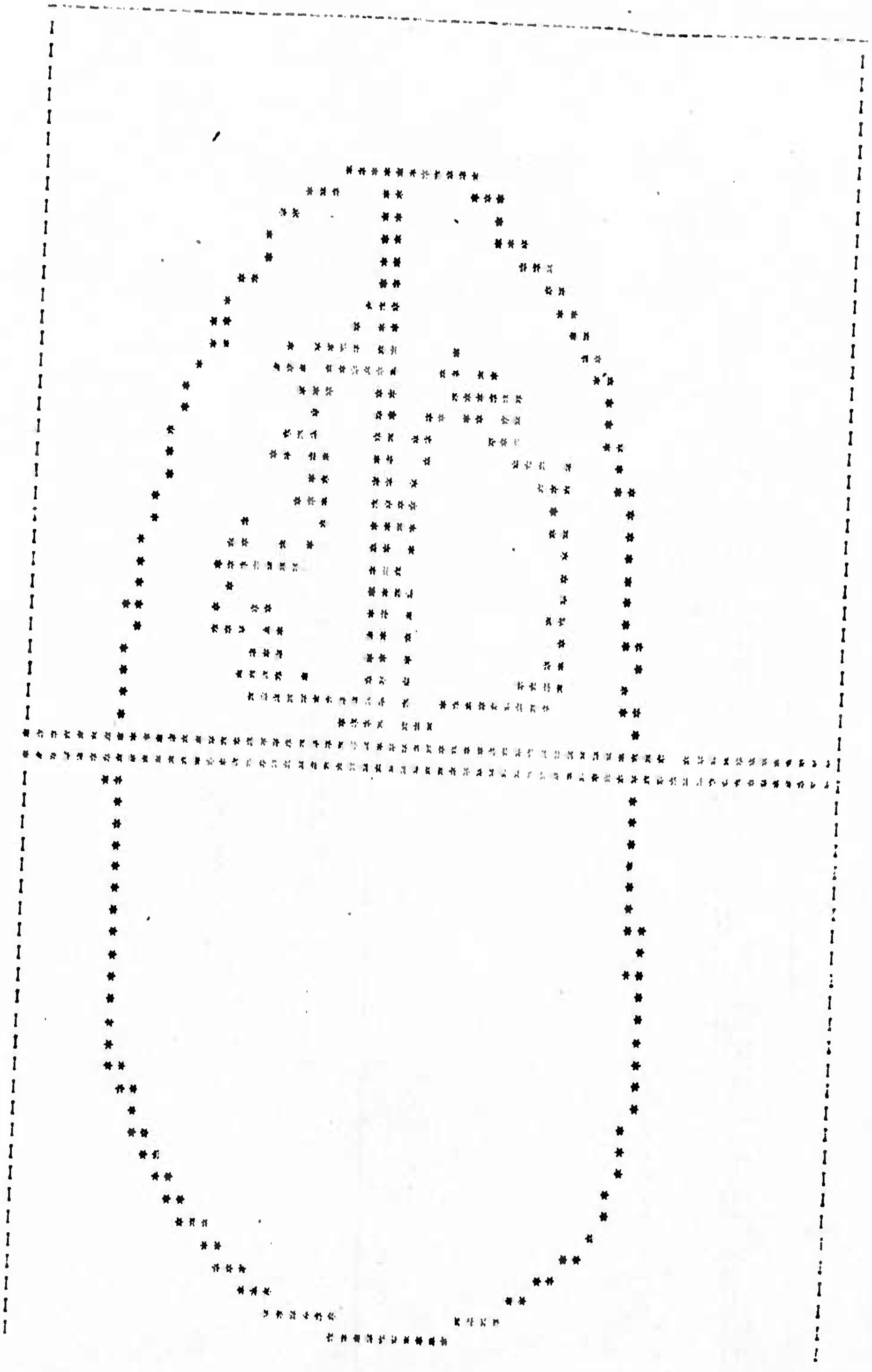


Fig. 13 Contours obtained from an unfiltered picture.

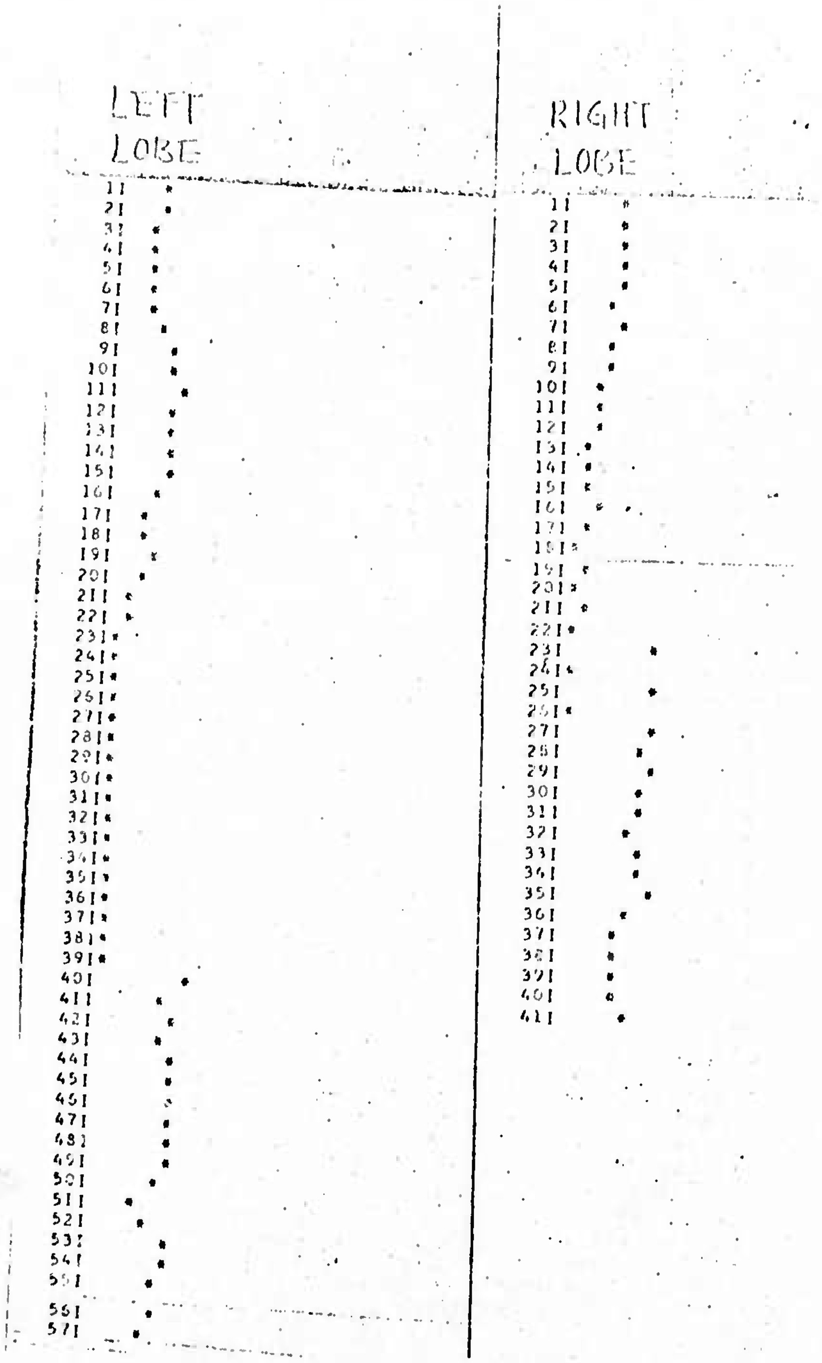


Fig. 14 Scatter plot.

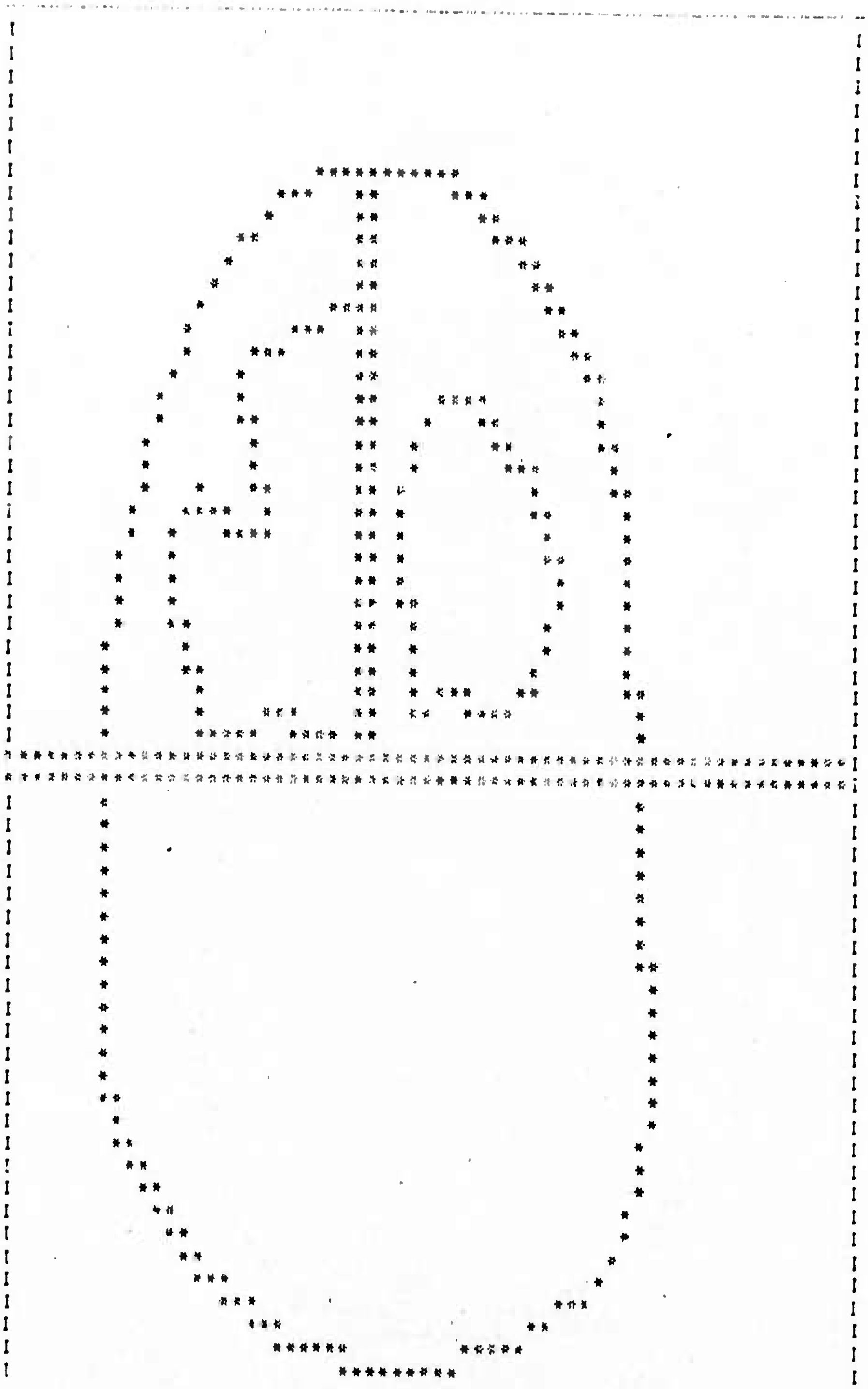


Fig. 15 P contours obtained with T=8000
PER 1 = 0.6, PER 2 = 0.4.

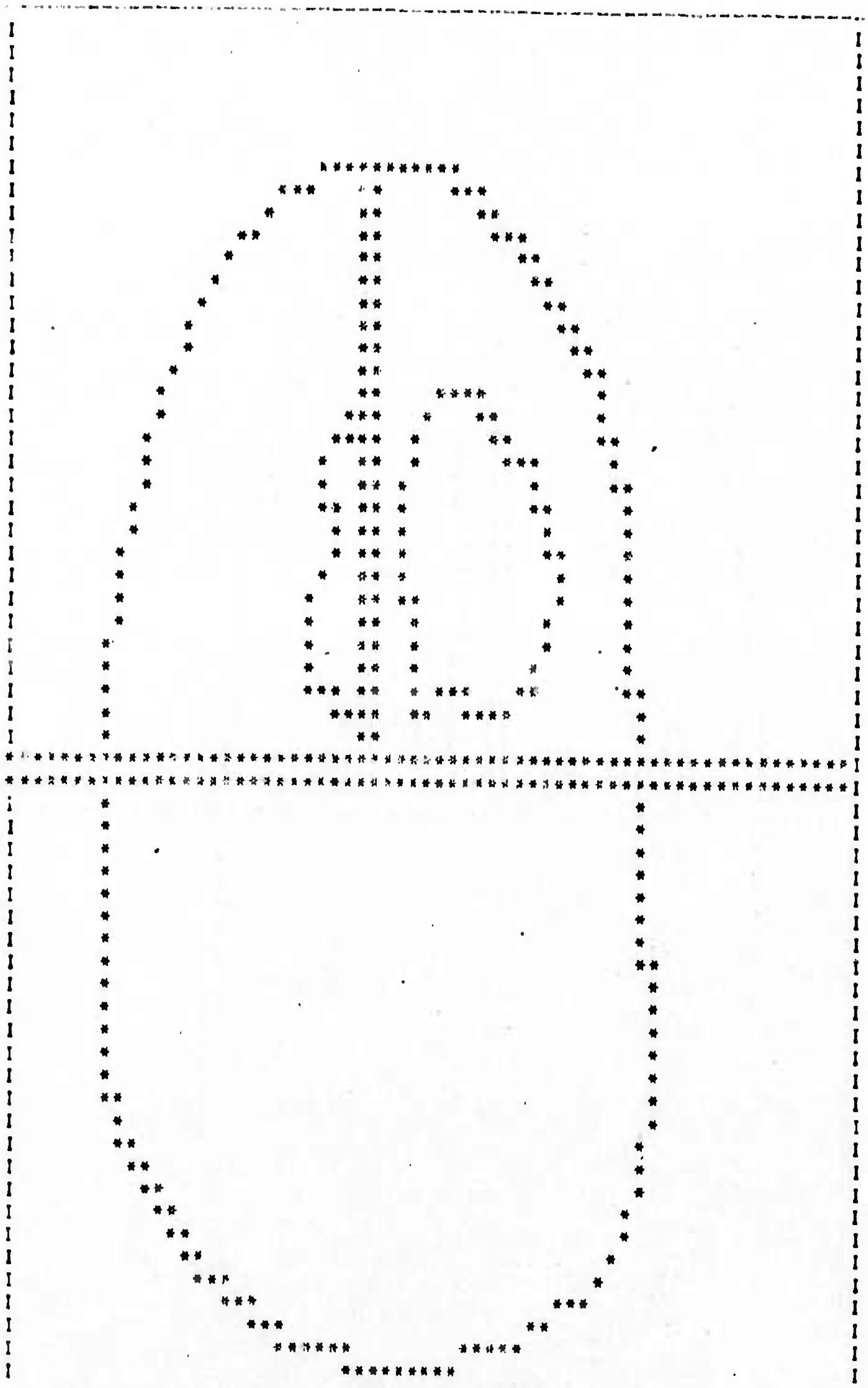


Fig. 16 Contours obtained with T=8000, PERl = 0.15, PER 2 = 0.4.

Table 1

LEFT LOBE

X	Y	direction code
15	25	4
16	24	4
17	24	3
18	24	3
19	24	3
20	24	3
21	24	3
22	23	4
22	22	5
22	21	5
21	20	6
21	19	5
21	18	5
21	17	5
21	16	5
22	15	4
23	15	3
24	15	3
25	14	4
26	14	3
27	15	2
28	16	2
28	17	1
28	18	1
28	19	1
28	20	1
28	21	1
28	22	1
28	23	1
28	24	1
28	25	1
28	26	1
28	27	1
28	28	1
28	29	1
28	30	1
28	31	1
28	32	1
28	33	1
27	34	8
26	33	6
25	33	7
24	32	6
23	32	7
22	32	7
21	32	7
20	32	7
19	32	7
18	32	7
17	31	6
18	30	4
18	29	5
17	29	7
16	29	7
15	28	6
14	27	6
14	26	5

RIGHT LOBE

X	Y	direction code
30	25	5
30	24	5
30	23	5
30	22	5
30	21	5
31	20	4
31	19	5
32	18	4
33	17	4
34	17	3
35	17	3
36	17	3
37	18	2
38	19	2
39	20	2
40	20	3
41	21	2
41	22	1
42	23	2
42	24	1
43	25	2
43	26	1
42	27	8
42	28	1
41	29	8
41	30	1
40	31	8
39	31	7
38	32	8
37	32	7
36	32	7
35	31	6
34	31	7
33	31	7
32	32	8
31	31	6
31	30	5
31	29	5
31	28	5
31	27	5
30	26	6

SATELLITE IMAGERY NOISE REMOVAL

O.R. Mitchell and P.L. Chen

I. Introduction.

This recent work is a continuation of the previous report "Satellite Imagery Noise Removal" [1]. We are using homomorphic filtering techniques to remove multiplicative noise effects such as cloud and atmospheric turbulence in ERTS imagery.

Our approach is to estimate the noise power spectrum by classifying each region of the noisy picture according to the level of the noise present using the multispectral data analysis software system developed by LARS. Once the noise power spectrum is obtained, an optimum filter is derived to separate the signal and noise. Information is then extracted from the recovered signal using multispectral classification techniques.

II. Theory of Cloud Removal.

Assume an image of the earth is produced when a light cloud cover exists over the region of interest as in Fig. 1.

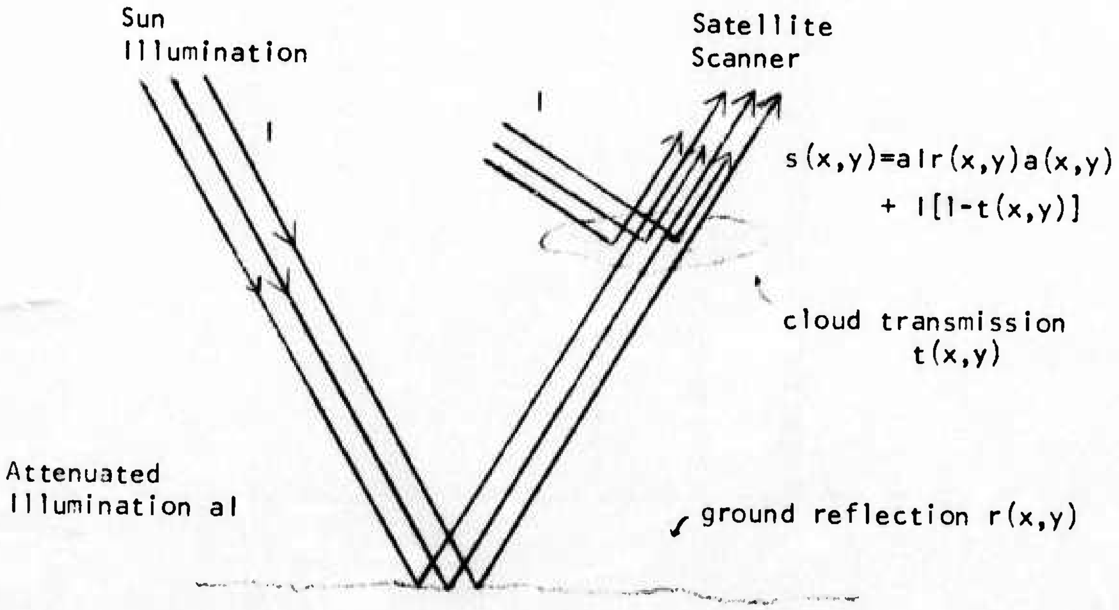

Sun
Illumination
I
Satellite
Scanner
I
$s(x,y)=aIr(x,y)a(x,y) + I[1-t(x,y)]$
cloud transmission $t(x,y)$
Attenuated
Illumination aI
ground reflection $r(x,y)$

Fig. 1 Satellite Scanner Image Components

We have assumed that the cloud reflection of sunlight plus the cloud absorption equals one and that the illumination is approximately constant on the earth surface. The image received at the scanner is

$$s(x,y) = aIr(x,y)\ t(x,y) + I[1-t(x,y)] \leq I$$

where $r(x,y)$ is the signal and $t(x,y)$ is the noise.

$$s(x,y) = It(x,y)\ [ar(x,y) -1] + I$$

Taking the logarith

$$\log[I-s(x,y)] = \log I + \log t(x,y) + \log[1-ar(x,y)]$$

Notice that now the signal and noise are additive so that an optimum linear filter can be applied.

The technique is as follows: "I" may be approximated as the brightest point in the image [corresponds to t=0]. The original data is therefore inverted by subtracting the intensity at each point from the maximum intensity in $s(x,y)$. The logarithm is then taken of the inverted data. Now the signal and noise are additive. The power spectrum of the noise $t(x,y)$ is estimated using techniques discussed in the next section and a Wiener filter is used to get the estimate of $\log[1-ar(x,y)]$. The signal is then recovered by taking the anti-log, inverting the intensities, and increasing the contrast.

III. The Classification of Cloud.

The analysis of clouds as a multiplicative noise in ERTS imagery is implemented by using the classification technique at LARS. In this report the classification of cloud is based on the amount of cloud cover located at each pixel.

The LARS classifier processes multispectral data one point at a time classifying unknown data using training statistics developed from pre-classified data.

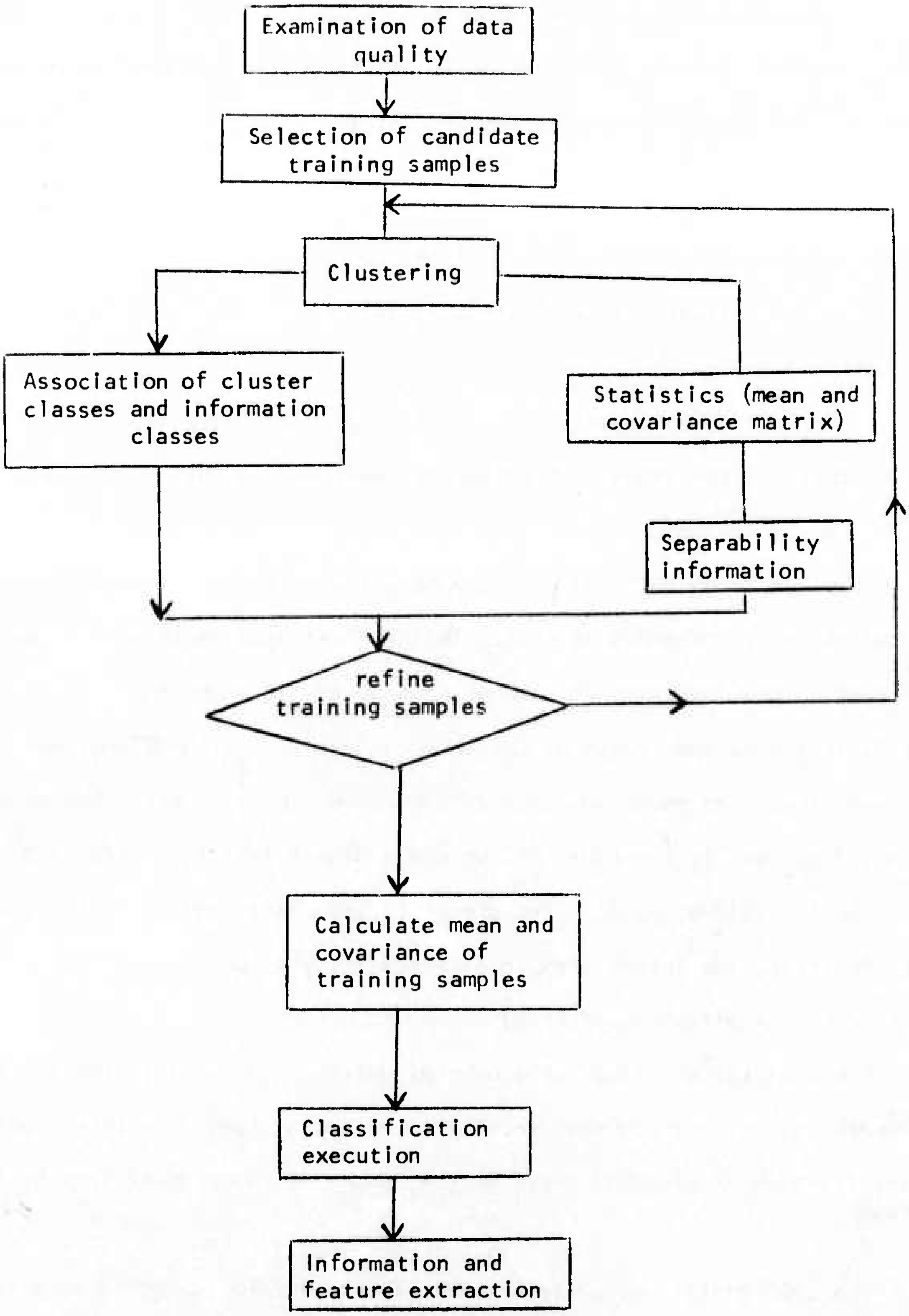

Examination of data quality
Selection of candidate training samples
Clustering
Association of cluster classes and information classes
Statistics (mean and covariance matrix)
Separability information
refine training samples
Calculate mean and covariance of training samples
Classification execution
Information and feature extraction

Fig. 2 Flow chart of LARS classification

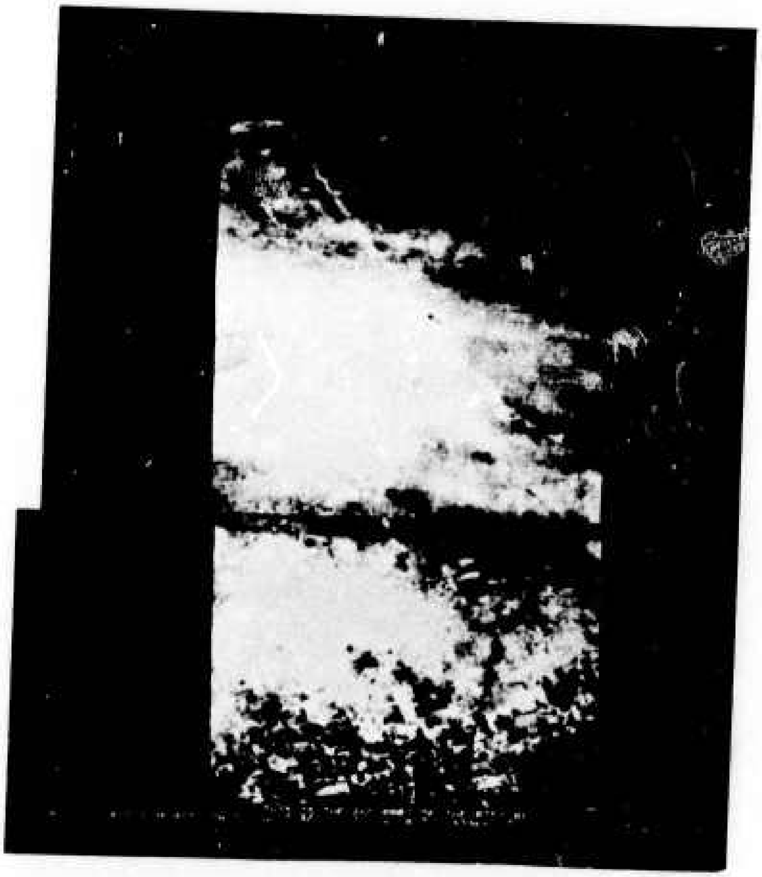


Fig. 3 A section of Chicago and vicinity
LARS Run 7300 3600
Channel 4

Fig. 4 Classified picture of Fig. 3

The basic assumptions for classification are as follows:

(1) All class probabilities are equal.

(2) The probability density function of training samples is Gaussian.

The decision criterion for classification is the so-called maximum likelihood rule. The unknown data point is compared with all of the training classes and then is assigned to the most likely class. The detailed information about classification algorithm can be found in LARS information notes [2,3].

A general procedure for the classification of clouds is described in the block diagram in Fig. 2.

Consider Fig. 3. This is a section of original ERTS-1 multispectral scanner data (channel 4). Obviously, a large number of classes of clouds can be defined, however, only four kinds are chosen. These are Full cloud (F), Most cloud (M), Half cloud (H), and Small cloud (•). Water and ground are defined to be a class of no cloud.

Fig. 4 shows a classified version of Fig. 3. It is hard to compare these two pictures, because the classified picture is based on all four channels.

The description of cloud as we defined in this manner has proved to be very successful when one sees the training and test field performance. Tables 2-1, 2-2 show these results.

IV. Noise Extraction.

Fig. 5 shows a small section of Fig. 4, which has size of 64x64 (line 838-964-2, Col. 1220-1346-2).

Notice that a classification map (Fig. 5) is simply different integers representing each class. Since these are six spectrally distinct classes, integers 1 through 6 are used to represent them. Fig. 6 shows this integer

Table 2-1 Training Field Performance

Group	No. of Samps	Pct. Corct	Number of Samples Classified Into					
			Fulcld	Moscld	Hahcld	Smlcld	Water	Ground
Fulcld	3	100.0	3	0	0	0	0	0
Moscld	2	100.0	0	2	0	0	0	0
Hafcld	6	100.0	0	0	6	0	0	0
Smlcld	15	100.0	0	0	0	15	0	0
Water	6	100.0	0	0	0	0	6	0
Ground	6	100.0	0	0	0	0	0	6
	38		3	2	6	15	6	6

Overall
Performance 38/ 38) = 100.0

(From LARS *PRINTRESULTS)

Table 2-2 Test Field Percentage (PCT.)

Group	No. of Samps	Percentage of samples classified into					
		Fulcld	Moscld	Hafcld	Smlcld	Water	Ground
Fulcld	10	100.0	0.0	0.0	0.0	0.0	0.0
Moscld	2	100.0	0.0	0.0	0.0	0.0	0.0
Hafcld	10	0.0	0.0	80.0	0.0	0.0	20.0
Smlcld	36	0.0	0.0	0.0	100.0	0.0	0.0
Water	8	0.0	0.0	0.0	0.0	100.0	0.0
Ground	6	0.0	0.0	0.0	0.0	0.0	100.0
	72	16.7	0.0	11.1	50.0	11.1	11.1

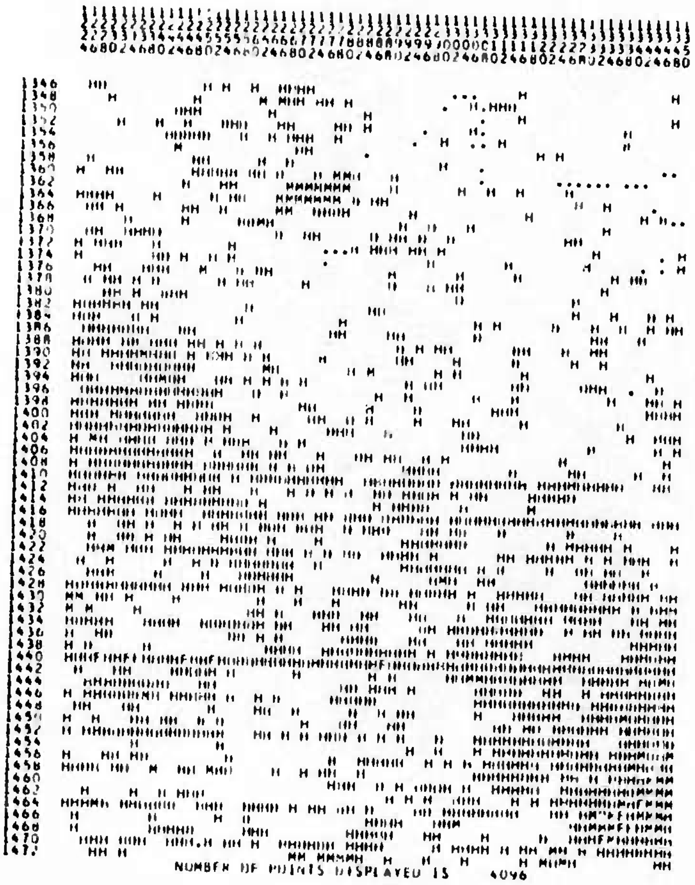
Overall
Performance 68/ 72) = 94.4

(From LARS *PRINTRESULTS)

CLASSES

SYMBOL	CLASS
F	FULCLD
M	MOSCLD
H	HAFCLD

SYMBOL	CLASS
.	SMLCLD
W	WATER
	GROUND

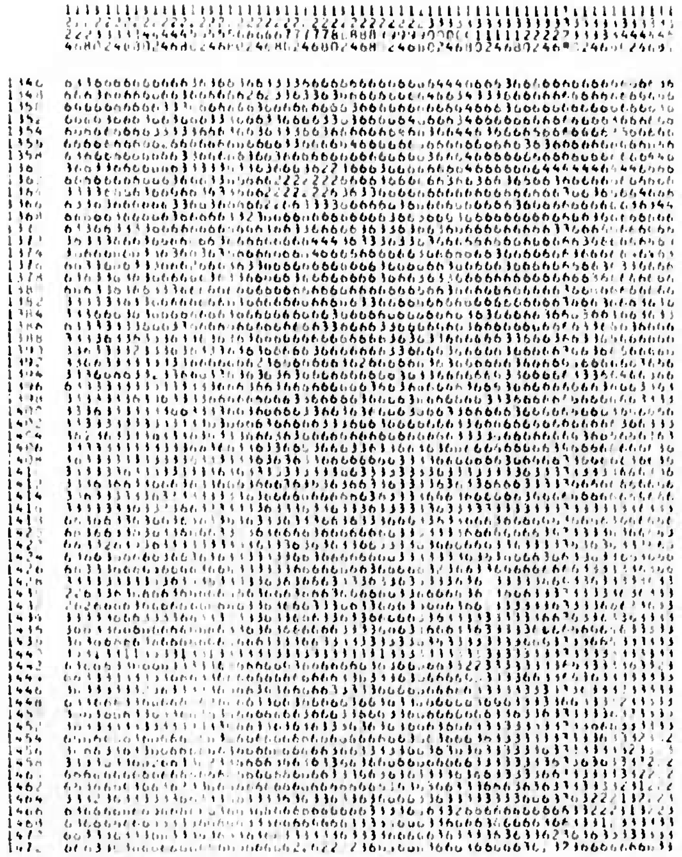


NUMBER OF POINTS DISPLAYED IS 4096

Fig. 5 Classification map of size 64x64. (*Lars Run 73003600, Line 1346-1472-2, Col. 1224-1350-2).

CLASSES

SYMBOL	CLASS	SYMBOL	CLASS
1	FOLCED	4	SMECED
2	MUSCLED	5	WATER
3	HAFCED	6	GROUND



NUMBER OF POINTS DISPLAYED IS 4096

Fig. 6 Integer coded classification map of Fig. 4.

classified map. A reasonable guess about the percentage transmission of reflected energy passing through the cloud is listed in the following table.

Table 3-1 Cloud blocking information.

Type of Cloud	Percentage of Transmission
Full Cloud	10%
Most Cloud	30%
Half Cloud	50%
Small Cloud	75%
Water	100%
Ground	100%

Consider Fig. 6 and Table 3-1 together, a pattern for noise only can be constructed simply by transforming each coded integer into its corresponding percentage value. Fig. 7 shows the picture for the noise only.

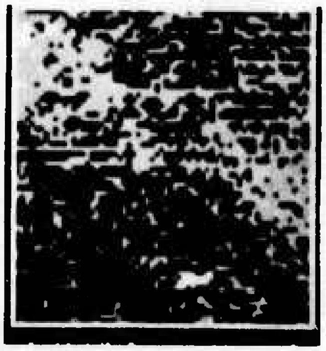


Fig. 7 Picture of cloud (Half and Small) and no cloud (Water and Ground).

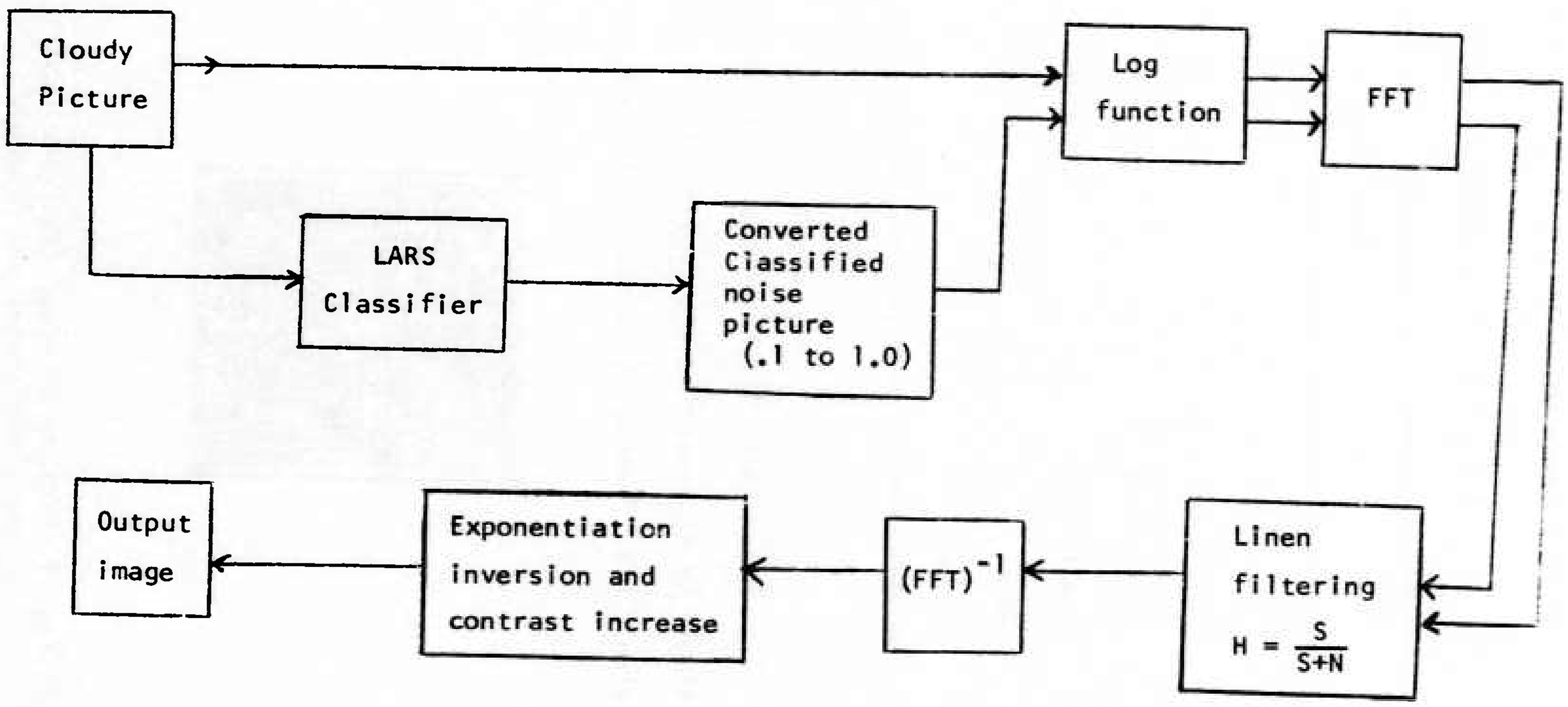

Cloudy Picture
LARS Classifier
Converted Classified noise picture (.1 to 1.0)
Log function
FFT
Linen filtering
H = S/S+N
(FFT)^-1
Exponentiation inversion and contrast increase
Output image

Fig. 8 Block diagram of homomorphic image filtering technique for cloudy picture.

classified map. A reasonable guess about the percentage transmission of reflected energy passing through the cloud is listed in the following table.

Table 3-1 Cloud blocking information.

Type of Cloud	Percentage of Transmission
Full Cloud	10%
Most Cloud	30%
Half Cloud	50%
Small Cloud	75%
Water	100%
Ground	100%

Consider Fig. 6 and Table 3-1 together, a pattern for noise only can be constructed simply by transforming each coded integer into its corresponding percentage value. Fig. 7 shows the picture for the noise only.

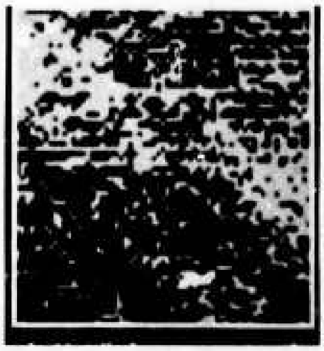


Fig. 7 Picture of cloud (Half and Small) and no cloud (Water and Ground).

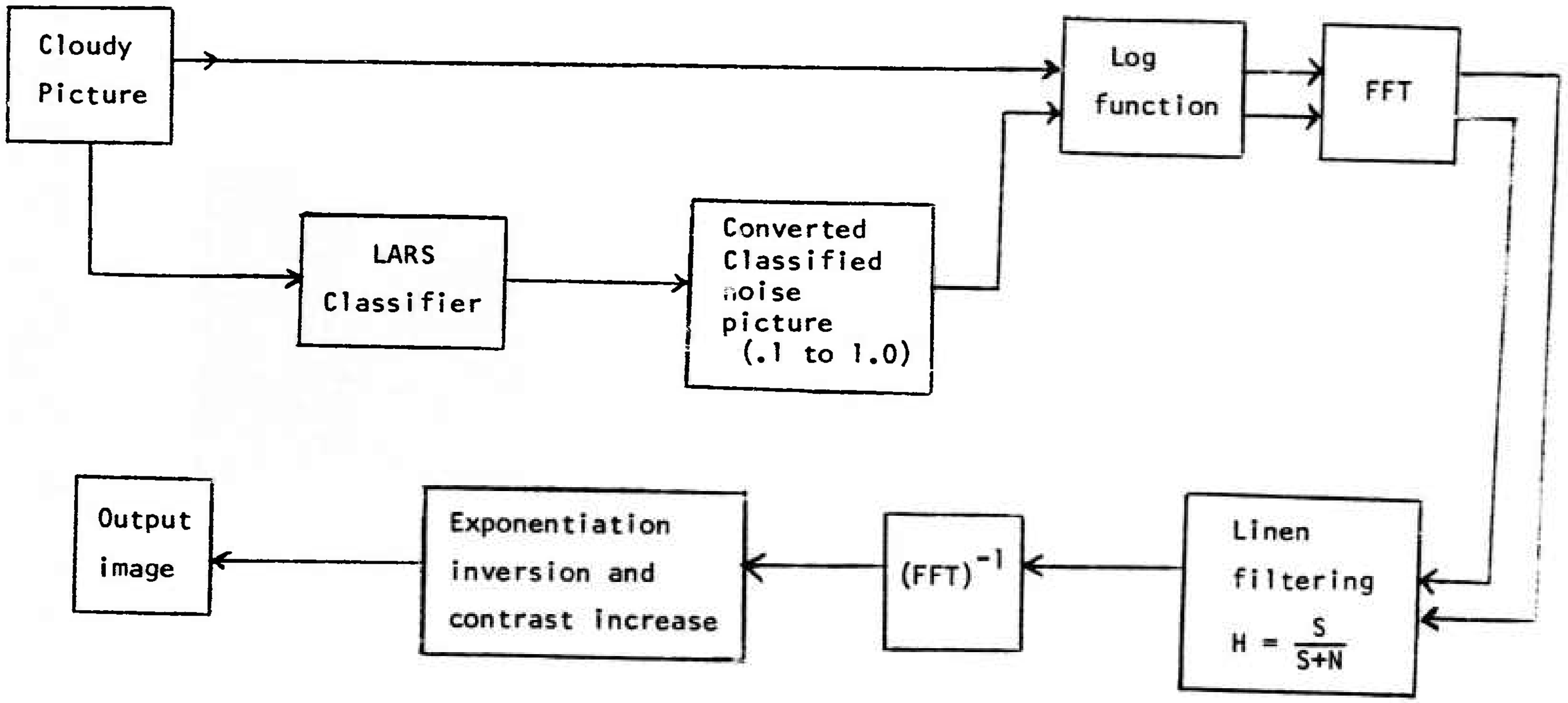


Fig. 8 Block diagram of homomorphic image filtering technique for cloudy picture.

V. Homomorphic Filtering Process.

The two dimensional filtering to reduce the cloud effect is accomplished by weighting each point in the discrete Fourier Transform of the logarithm of the picture grey levels by a non casual filter function of the form $\frac{S_i}{S_i+N_i}$ where S_i and N_i are the vlaues of the signal and noise power spectrums at the particular frequency point being considered. Following the filtering process we take the inverse DFT and then exponentiate to obtain the filtered picture. See Fig. 8.

VI. Results and Conclusions.

The filter function tends to smooth the picture resulting in a loss of sharp edges. However, the synthesis of high-passed picture to filtered picture will compensate this disadvantage.

The overall performance of the filtering processes, described in this report, will be evaluated by classifying the filtered picture and comparing with the ground true map of the same area. This will be done in the near future. We are also converting the filtering operation to a three dimensional filter, to take advantage of the four spectral channels in the filter.

REFERENCES

[1] O.R. Mitchell and P.L. Chen, "Satellite image noise removal," Quarterly Report under ARPA for period February 1, 1975 - April 30, 1975.

[2] J.C. Lindenlaub, "Guide to multispectral data analysis," The Laboratory for Application of Remote Sensing, Purdue University, 1973.

[3] T.K. Cary and J.C. Lindenlaub, "A case study using LARSYS for analysis of LANDSAT data," The Laboratory for Application of Remote Sensing, Purdue University, 1975.

References

[1] O.R. Mitchell and P.L. Chen, "Satellite image noise removal," Quarterly Report under ARPA for period February 1, 1975 - April 30, 1975.

[2] J.C. Lindenlaub, "Guide to multispectral data analysis," The Laboratory for Application of Remote Sensing, Purdue University, 1973.

[3] T.K. Cary and J.C. Lindenlaub, "A case study using LARSYS for analysis of LANDSAT data," The Laboratory for Application of Remote Sensing, Purdue University, 1975.

PDP-11 SOFTWARE DEVELOPMENTS

James J. Besemer

The major accomplishment on the PDP-11 this year was the installation of a new operating system called UNIX. It was developed at Bell Labs, and is distributed by Western Electric. The University of Illinois engineered the system modifications to allow it to communicate with the ARPANET. Our version of the system has been fully operational since early October, and has been operating on a 24 hour per day schedule since the first of the year. We are quite pleased with the new system. With the availability of UNIX, most of the graduate students and other researchers have been working on transporting their software from other sites to our PDP-11. Other accomplishments include the (earlier) installation of an ELF Telnet program, which allowed us our first access to the ARPANET, and the development of other supporting software.

In July, with cooperation from the staff at the ILLIAC-IV Center for Advanced Computation, our first running ELF TELNET server was installed. It is a copy of the ELF software, custom tailored to our system. The current version supports TELNET through our terminals and FTP to our line-printer. Since we expected to replace the system with NETWORK-UNIX, we did not expand the ELF to use the other devices in our system. Since UNIX did not have the network software installed when it first arrived, ELF served as our primary entry to the network. However, when the full, NETWORK-UNIX system was operational, ELF became obsolete, and is rearly used, except in those instances where hardware problems disallow the use of UNIX, yet still allow ELF to run.

Our use of the ARPANET remains small. Most of our research has been carried out either on our own PDP system, or at the Laboratory for the Application of Remote Sensing and the Purdue campus central facility. These latter sites have the large, scientific machines required for our research (two CDC 6600's and an IBM 360). In the future, we plan to use the 360/91 at UCLA and the ILLIAC-IV.

The development of PDP-11 software over the year can be considered quarter by quarter. During the first quarter our primary efforts were directed towards checking out the hardware and getting the DOS system operational. This was complicated by the fact that there were many hardware problems, and by the fact that the DOS system did not arrive until the middle of January. This delayed the correction of the hardware problems because many of them were not discovered until after DOS was delivered. We also spent time the first quarter familiarizing ourselves with the operation of the equipment and the DOS software. During the second quarter we concerned ourselves with developing basic software for DOS which allowed access to the non-standard devices, like the COMTAL color display and the VERSATEK electrostatic printer, and other basic software to use these devices. We also were investigating the various network software products which were available at that time. We were considering both ANTS and ELF. By the end of the second quarter, we had enough information about ANTS and ELF, and enough experience with the ARPANET to be able to begin working on the installation of a network software system. Most of the third quarter was spent first by debugging the IMP-interface hardware, and working on getting ELF running on our machine. The software portion of this work was complicated by the fact that we did not have access to the network except via a noisy line to Illinois. Our original plan was to get a magnetic tape containing the ELF sources, and compile them on our PDP-11, using DOS. Unfortunately, the DOS system was not able to compile the (very large) ELF sources, and this was not discovered until several abortive attempts were made. About the time that the hardware was finished (late May), we accepted an offer from the ILLIAC-IV IAC to help us get a copy of their ELF running on our machine. With their assistance, we got the network software running in early July. It was about this time when we heard about UNIX, and the network software developed for it by University of Illinois. We researched it, and decided that for a machine as large as ours, that ELF wasted most of the equipment we had. UNIX, on the other hand, allowed us to access the network, and to perform local processing, simultaneously. Thus most of the last quarter was spent installing, improving and testing the UNIX system. We were very pleased. UNIX, used only for network access turned out to be better than ELF, and used only for local processing was better than DOS or RSX. Taken as a whole, it seems to be the best system available for a moderately sized system like ours. This opinion seems to be shared by other PDP-11 network users, as several of the old ELF-followers have switched to NETWORK-UNIX. UNIX supports FORTRAN, PDP-11 assemply language, and a high-level language, similar to PL/1, called "C". It also supports several "compiler-compilers" and has a computer language translator, which will be a tremendous aid to us as we translate out CDC and IBM software to run on the PDP-11. That UNIX is an easy system to use is supported by the fact that to develop the software to run our COMTAL display and the electrostatic printer has only about 20% of the development time for the equivalent software for DOS.

This next year we are continuing the development of more UNIX support software, and converting old programs to run under UNIX; and in particular, we are starting to do most of our new research projects on the PDP-11 UNIX system.

Current Equipment Configuration
February, 1976

QTY	Manufacturer	Description
3	Beehive Elect.	"Super-Bee" Terminals
2	Tex. Inst.	"Silent 700" Terminals
1	Digi-Data	Industry standard magnetic tape system; 2, 9-track and 1, 7-track drives; one each NRZI and phase-encoded formatters/controllers
1	DEC	Dual-drive DECtape unit
1	DEC	RP03 disk drive (40 million characters)
1	Fabritek	96K-word auxiliary memory system (64K bought by ARPA, 32K by NASA)
1	Versatek	Electrostatic matrix printer
1	Comtal	Color picture display
1	Data Printer	132 column, 600 L.P.M. line printer
1	True-Data	Punched card reader
1	Tektronix	Model 4010, graphics display
1	DEC	PDP-11/45 computer system; system includes: 32K memory FPP-11 floating point processor (NSF money) H960 extension mounting cabinet 3 - small peripheral mountings blocks (DD-11) 1 UNIBUS repeater/expander DH11, 16-line terminal multiplexor KW11-p programmed clock "ANTS" - type PDP-11/IMP interface

BOOKS

HUANG, T., Image Processing and Digital Filtering (editor), Springer-Verlag, 1975.

KAK, A., Digital Picture Processing, Academic Press, New York, Fall 1975, (with A. Rosenfeld).

JOURNAL PUBLICATIONS

FUKUNAGA, K., "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition," IEEE Trans. on Information Theory, Vol. IT-21, pp. 32-40, January 1975, (with L.D. Hostetler).

FUKUNAGA, K., "k-Nearest Neighbor Bayes-Risk Estimation," IEEE Trans. on Information Theory, Vol. IT-21, pp. 258-293, May 1975, (with L.D. Hostetler).

FUKUNAGA, K., "A Branch and Bound Algorithm for Computing k-Nearest Neighbors," IEEE Trans. on Computers, Vol. C-24, pp. 750-753, July 1975, (with P.M. Narendra).

FUKUNAGA, K., "A Branch and Bound Clustering Algorithm," IEEE Trans. on Computers, Vol. C-24, pp. 908-915, September 1975, (with W.L.G. Koontz and P.M. Narendra).

HUANG, T., "Bounds on Bit Rates of Linear Runlength Codes," IEEE Trans. on Inf. Theo., November 1974.

HUANG, T., "Iterative Image Restoration," Applied Optics, Vol. 14, No. 5, pp. 1165-1168, May 1975, (with B. Baker and S. Berger).

HUANG, T., "Coding of Multilevel Graphics," IEEE Trans. on Communications, Vol. 23, No. 6, pp. 598-606, June 1975, (with H. Koller).

HUANG, T., "Image Restoration by Singular Value Decomposition," Applied Optics, September 1975, (with P. Narendra).

HUANG, T., "The Importance of Phase in Image Processing Filters," IEEE Trans. on Acoustics, Speech, and Signal Processing, December 1975, (with J. Burnett and A. Deczky).

HUANG, T., "Facsimile Coding by Skipping White," IEEE Trans. on Comm., December 1975, (with A. Hussian).

HWANG, K., "A TTL Programmable Logic Array and Its Application," IEEE Proceedings, accepted to appear as a proceedings letter, 1975.

HWANG, K., "Some Structural Complexities of Time-Varying Sequential Machines," International Journal of Computers and Information Sciences, September 1975.

KAK, A.C., "Signal Processing in the Determination of Acoustic Impedance Profiles," TR-EE 75-7, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907, (with L.R. Beaumont and J. Wolfley).

PANDA, D.P., "Pattern Recognition Applied to Surgery," Computers in Biology and Medicine, Vol. 4, 1975, (with E.A. Patrick, F. Stelmack, and S. Jardina).

WINTZ, P.A., "On the Effect of Timing Errors in Run Length Codes," IEEE Trans. on Communications, June 1975, (with L.C. Wilkins).

WINTZ, P.A., "A Boundary Finding Algorithm and its Applications," IEEE Trans. on Circuits and Systems, March 1975, (with J. Gupta).

CONFERENCES

HUANG, T., "Facsimile Coding by Skipping White," presented at the 1976 Picture Coding Symposium, Asilomar, January 28-30, 1976.

HUANG, T., "Coding of Two-Tone Images," presented at the SPIE Seminar on Picture Coding, San Diego, August 21-22, 1975.

HUANG, T., "Application of Edge Detection to Image Enhancement," presented at the 5th Annual Symp. on Automatic Imagery Pattern Recognition, University of Maryland, College Park, MD, April 9-11, 1975, (with G. Tang).

HUANG, T., "Optical/Digital Information Processing," Panel discussion, Int'l Conf. on Optical Computing, Washington, D.C., April 23-25, 1975.

HUANG, T., "Electronic Processing in Industrial Quality Control," Proc. SPIE seminar on Quality Control and Reliability, San Diego, CA, May 15-16, 1975.

HUANG, T., "Easily Implemental Suboptimum Runlength Codes," Proc. Int'l Comm. Conf., San Francisco, CA, June 16-18, 1975.

HUANG, T., "Restoration of Images Degraded by Spatially-Varying Systems," to be presented at the OSA Technical Meeting on Image Processing, Asilomar, February 24-26, 1976.

HUANG, T., "Nonlinear Estimation of Markov Jump Processes," to be presented at the IEEE Int'l Information Theory Symposium, Ronneby, Sweden, June 21-24, 1976, (with J. Burnett).

HUANG, T., organizer and lecturer, "Image Coding, Enhancement, and Recognition," UCLA, March 3-7, 1975.

HUANG, T., organizer and lecturer, "Digital Techniques in Spectral Analysis, Estimation, and Filtering," a Computer Workshop, Purdue University, March 10-14, 1975.

HUANG, T., organizer and lecturer, "Digital Filtering - Applications to Speech and Image Processing," UCLA, May 5-9, 1975.

HUANG, T. and KAK, A.C., organizers and lecturers, "Digital Image Processing," Institute for Image Technology, Silver Spring, MD, November 10-14, 1975.

HUANG, T., organizer and lecturer, "Charge-Coupled and Surface Acoustical Wave Devices," Institute for Image Technology, West Lafayette, Indiana, October 6-10, 1975.

HWANG, K., "Parallel Processing of Multiway Search Trees," Vol. 981-10, pp. 170-76, Proc. of the Conference on Computer Graphics, Pattern Recognition, and Data Structure, at Beverly Hills, CA, May 14-16, 1975, (with S.B. Yao).

HWANG, K., "Cyclic Stochastic Sequential Systems," Department of Statistical Sciences, Purdue University, March 1975.

HWANG, K., "Parallel Internal Searching in Multiprocessor Computers," University of Notre Dame, March 1975.

KAK, A.C., "Noniterative Impedance Profile - Impulse Response Relationships and Signal Processing," presented at Seminar on Ultrasonic Tissue Characterization, Gaithersburg, MD, May 28-30, 1975.

KAK, A.C. and HUANG, T., organizers and lecturers, "Novel Imaging Techniques in Medical and Industrial Applications," Institute for Image Technology, West Lafayette, Indiana, September 29-October 3, 1975.

KAK, A.C., "Algorithms for Reconstruction," presented at the International Meeting on Cardiovascular Imaging and Image Processing, Stanford University, July 1975.

KAK, A.C., "Computerized Tomography Using Video Recorded Flouroscopic Images," presented at the Conference on 2-D and 3-D Image Reconstruction from Projections, Stanford University, August 1975, (with C. V. Jakowatz and N. A. Baily).

MITCHELL, O.R., "Information Extraction Techniques for Drug Related Effects on the Evoked EEG," presented at the Midwest Biomedical Engineering Conference and Workshop, Columbus, OH, April 11-12, 1975, (with W.P. Jones).

MITCHELL, O.R., "Statistical Measures for the Determination of Texture Primitives," presented at the Fifth Annual Symposium on Imagery Pattern Recognition, College Park, MD, April 17-18, 1975, (with C.R. Myers).

WINTZ, P.A., "TV Bandwidth Compression," (invited), Proc. of the SPIE 19th Annual Technical Symposium, August 1975, (with J. Wilson and E. Schmidt).

WINTZ, P.A., "Image Filtering, Coding, and Information Extraction," Seminar presented at Stanford University, Stanford, CA, April 16, 1975.

WINTZ, P.A., "Two-Dimensional Low Pass Filters for Images," Seminar presented at University of California, Los Alamos Research Laboratory, Los Alamos, NM, April 10, 1975.

WINTZ, P.A., "Earth Resources Data Processing," Seminar presented at the University of Tennessee, Knoxville, TN, November 1974.

WINTZ, P.A., organizer and lecturer, Wintek Image Processing Short Course, Lafayette, IN, May 13, 1975.

WINTZ, P.A., "Purdue Image Processing," presented at IEEE Conference on Decision and Control, Phoenix, Arizona, November 1974.

WINTZ, P.A., "Image Pattern Recognition for Satellite On-Board Processing of ERTS Resource Data," presented at Pattern Recognition Workshop co-sponsored by NSF and Government Products Div., EIA University of Maryland, Silver Spring, MD, November 1974.

WINTZ, P.A., organizer and lecturer, Technology Service Corporation - Wintek Corporation, (co-sponsored), Image Processing Short Course, Zurich, Switzerland, March 17-21, 1975.

WINTZ, P.A., organizer and lecturer, Wintek Image Processing Short Course, Palo Alto, CA, April 15, 1975.

STAFF

CO-PRINCIPAL INVESTIGATORS

P. Wintz
T. Huang

PROFESSORIAL

G. Cooper
K. Fukunaga
K. Hwang
A. Kak
O. Mitchell

POSTDOCTORAL RESEARCHERS

M. Kaveh

GRADUATE RESEARCHERS

S. Berger
W. A. Boyne
J. Burnett
D. Chan
Wm. Chan
P. H. Chen
P. L. Chen
R. Clark
C. Myers
P. Narendra
B. O'Connor
D. Panda
A. Salahi
G. Sherman
L. Stanfield
G. Tang
E. Wiswell
M. Yoo

RESEARCH STAFF

J. Besemer
W. Robey

UNDERGRADUATE RESEARCHERS

C. Buckwacter
M. DeMoney
R. Johnson
P. Miller
J. Schwab
J. Uban

ELECTRONIC TECHNICIANS

D. Azpell

SECRETARIES

M. Barbour
M. Claire

MISSION
of
Rome Air Development Center

RADC is the principal AFSC organization charged with planning and executing the USAF exploratory and advanced development programs for information sciences, intelligence, command, control and communications technology, products and services oriented to the needs of the USAF. Primary RADC mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, and electronic reliability, maintainability and compatibility. RADC has mission responsibility as assigned by AFSC for demonstration and acquisition of selected subsystems and systems in the intelligence, mapping, charting, command, control and communications areas.

AMERICAN REVOLUTION BICENTENNIAL 1776-1976